

ML11

ML11 LOGIC TEST
CZMLABO

AH-S390B MC
FICHE 1 OF 3

MAY 98 I
COPYR GATE 98
MADE IN USA



Table with multiple columns and rows of data, likely representing logic test results. The content is extremely faint and illegible due to the low contrast of the scan. The table appears to have approximately 15 columns and 20 rows of data points.

ML11

ML11 LOGIC TEST
CZMLABO

AH S390B MC
FICHE 2 OF 3

MAY 98
COPYR 98
MADE IN USA



The main body of the document is a large grid of approximately 20 columns and 20 rows of small, illegible text blocks. Each block appears to be a test case or a data entry point, possibly related to logic testing as indicated by the header. The text is too small to be transcribed accurately.

ML11

ML11 LOGIC TEST
CZMLABO

AH S390B MC
FICHE 3 OF 3

MAY 1981
COPYRIGHT 1981
MADE IN USA



A microfiche card containing a grid of 12 columns and 12 rows of data. Each cell in the grid contains a small, dense block of text, likely representing a single test result or data point. The text is too small to be legible in this image.



1
2

.TITLE CZMLABO ML-11 LOGIC TEST
.SBTTL USER DOCUMENTATION
.REM 2

IDENTIFICATION

PRODUCT CODE: AC-S388B-MC
PRODUCT NAME: CZMLABO 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
4-	1	PROGRAM HEADER AND TABLES
43-	1	MISCELLANEOUS CODING SECTION
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 HIERARCHY 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 (CHOUS).

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.

<u>COMMAND</u>	<u>EFFECT</u>
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:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = 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
IBE*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXE*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
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)
IDU	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 DIALOGE

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

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 (D) ? 160000<CR>
SUB-DEVICE # (D) ? 0<CR>
Q-FACTOR (D) 0 ? 1<CR>

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

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

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

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

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

UNIT 7
CSR ADDRESS (D) ? 160000<CR>
SUB-DEVICE # (D) ? 6<CR>

Q-FACTOR (0) 0 ? 1<CR>

UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 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 (0) ? 8<CR>

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

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

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 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 (0) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0-7<CR>
Q-FACTOR (0) 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:XX\XXX
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

<u>ERROR NO.</u>	<u>FAILING LOGIC</u>
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

99 FAILURE DURING NON LAST
BLOCK TRANSFERS.

100 LBT BIT NOT CLEAR
AFTER LOADING DSA REG

101 LBT BIT NOT SET
AFTER A LAST BLOCK
TRANSFER

102 DSA REGISTER
INCREMENT FAILURE
AFTER A LAST BLOCK
TRANSFER

103 IAE BIT NOT SET AT
INVALID SECTOR ADDRESSES

104 AOE BIT NOT SET
AFTER ADDRESS OVERFLOW

105 SC BIT NOT AFTER
CS1 FUNCTION ADORT

106 GOOD DATA BUS
PARITY NOT DETECTED

107 GOOD DATA BUS
PARITY NOT GENERATED

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

109 UNS BIT NOT SET WITH
BAD UV DATA

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

111 UNIQUE REGISTER
SELECTION TEST FAILURE

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

113 UNIQUE ARRAY MODULE
ROW/COL ADDRESSING
FAILURE

114 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)
- 118 DATA DIAGNOSTIC REGISTER D1 D2 E2
INITIALIZATION ERRORS
- 119 RH CONTROLLER FAILURE TO RESPOND
TO EXISTANCE PROBE
- 120 NED BIT SET DURING MASS BUS TO
UNIBUS COMMUNICATION PROBE
- 121 DATA DIAGNOSTIC REGISTER D1 D2 E2
ONE'S / ZEROE'S READ WRITE ERRORS
- 122 DATA DIAGNOSTIC REGISTER D1 D2 E2
SHIFTING 1'S AND 0'S READ WRITE ERRORS
- 123 ECC HARD ERROR BIT NOT SET WHEN
UNCORRECTABLE ECC ERRORS WERE READ
- 124 ECC ERROR REGISTER FAILURE TO LATCH
FAILING ERROR CORRECTION INFORMATION
- 125 ECC ERROR REGISTER FAILURE TO CLEAR
- 126 ECC ERROR LOCATION REGISTER FAILED
TO CLOCK IN DSA ADDRESS
- 127 ECC ERROR LOCATION REGISTER DATA
BIT ERRORS
- 128 ECC ERROR LOCATION REGISTER
INITIALIZATION ERRORS
- 129 ECC HARD ERROR BIT FAILED TO SET
NOT SET
- 130 DATA CHECK ERROR BIT FAILED TO
SET / NOT SET
- 131 CRC DATA BUS FAILURE.
BIT CONTINUITY / UNIQUENESS ERRORS

- 132 UNIQUE NIBBLE CRC GENERATION CODE FAILURES
- 133 UNIQUE WORD CRC GENERATION CODE FAILURES
- 134 CORRECTABLE ERROR SYNDROME DECODE FAILURE. FAILURE TO COMPLIMENT EXPECTED FAILING BIT(S)
- 135 CORRECTABLE ERROR SYNDROME DECODE FAILURE. UNEXPECTED BIT(S) FOUND COMPLIMENTED
- 136 ECH OR UNC BIT NOT SET DURING READS WITH ECC ERRORS IN CHANNEL > 35
- 137 ERROR CORRECTION WAS NOT INHIBITED DURING UNCORRECTABLE ECC ERRORS. UNEXPECTED BIT(S) WERE FOUND COMPLIMENTED
- 138 ECH OR UNC BIT NOT SET WHEN MULTIPLE CHANNEL ERRORS WERE READ
- 139 ERROR CORRECTION WAS NOT INHIBITED DURING UNCORRECTABLE ECC ERROR. UNEXPECTED BIT(S) WERE FOUND COMPLIMENTED
- 140 ECC FAILURE TO DETECT AND CORRECT SINGLE BIT CHANNEL ERRORS
- 141 ECC FAILURE TO DETECT AND CORRECT MULTIPLE BIT CHANNEL ERRORS

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 DRIVE 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. MASS BUS READY
TEST THE RH CONTROLLER FOR EXISTANCE
- TST2. MASS BUS HANDSHAKE
TEST MASS BUS ---> UNIBUS COMMUNICATIONS
- TST3. DRIVE PRESENT
TEST TO SEE IF THE DRIVE UNDER TEST EXIST.
- TST4. DRIVE SELECTION
SEE IF SELECTING OTHER DRIVES ON RH EFFECTS DRIVE UNDER TEST.
- TST5. READ WRITE REGISTER ONE'S ZERO'S TEST
TEST REGISTERS READ WRITE CAPABILITY AND UNIQUENESS
- TST6. READ WRITE REGISTER SHIFTING ONE'S AND ZEROES
TEST REGISTERS FOR UNIQUE DATA BITS.
- TST7. REGISTER INITIALIZATION
TEST REGISTERS FOR CORRECT INIT DATA.
- TST8 REGISTER SELECTION TEST
TEST FOR UNIQUE REGISTER SELECTION
- TST9. PRINT DRIVE SERIAL NUMBER
PRINT THE CONTENTS OF MLSN IF THE SOFTWARE QUESTION WAS ANSWERED 'YES'.
- TST10. C-BUS PARITY
TEST IF DRIVE CAN DETECT BAD PARITY ON C-BUS AND GENERATE GOOD PARITY.
- TST11. MEMORY SIZING
SEE IF MEMORY SIZING LOGIC DETECTS AND RECORD CORRECT NUMBER OF ARRAYS PRESENT.
- TST12. NO-OP FUNCTION
SEE IF A NO-OP FUNCTION CAN BE EXECUTED.
- TST13. WRITE CHECK FUNCTION

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

TST14. WRITE FUNCTION

SEE IF A WRITE FUNCTION CAUSES THE DRIVE TO HANG.

TST15. READ FUNCTION

SEE IF A READ FUNCTION CAUSES THE DRIVE TO HANG.

TST16. CLEAR FUNCTION

SEE IF A CLEAR FUNCTION CAN BE EXECUTED.

TST17. DIAGNOSTIC REGISTER READ WRITE

TEST THE DIAGNOSTIC REGISTERS FOR
1'S/0'S, SHIFTING 1'S/0'S AND
INITIALIZATION

TST18. COMPOSITE ERROR BIT TEST

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

TST19. ATA BIT

TEST IF THE ATA BIT CAN BE SET AND CLEARED.

TST20. SEARCH FUNCTION

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

TST21. READ IN PRESET

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

TST22. ILLEGAL FUNCTION

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

TST23. 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.

TST24. INITIAL PROM TEST

TEST PROMS FOR EXISTENCE.

TST25. PROM 'OR' FUNCTION TEST

TEST THE PROM DATA ORING FUNC

TST26. UV ERROR TEST

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

TST27. INITIAL ARRAY TEST

TEST ARRAY TIMING AND CONTROL FOR EXISTENCE.

TST28. PROM SELECTION TEST

TEST FOR UNIQUE PROM SELECTION.

TST29. 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.

TST30. REFRESH TIMING

TEST TO SEE IF MEMORY CAN BE REFRESHED.

TST31. ADDRESS COUNTER

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

TST32. ARRAY MODULE SELECTION

TEST FOR UNIQUE ARRAY MODULE SELECTION

TST33. SEQUENCER EXISTENCE TEST

TEST TO SEE IF BASIC SEQUENCER TIMING EXISTS.

TST34. SYNC DATA BUS CONTINUITY/WRITE PATH

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

TST35. SYNC DATA BUS CONTINUITY/READ PATH

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

TST36. RAM-BUS ADDRESS COUNTER/WRITE PATH

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

TST37. RAM BUS ADRS COUNTER/READ PATH

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

- TST38. 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.
- TST39. 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
- TST40. ARRAY ADDRESS MUX
TEST FOR UNIQUE ROW AND COLUMN ADDRESSING
- TST41. NIBBLE OFFSET
TEST NIBBLE OFFSET COUNTERS TO COUNT TO 14 NIBBLE DATA TO BE SHIFTED ON DETECTION OF BAD NIBBLES.
- TST42. 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.
- TST43. LAST BLOCK INDICATOR
TEST THE LAST BLOCK INDICATOR BIT FOR NOT SETTING BELOW THE LAST AND SETTING AND CLRING AT THE LAST BLOCK
- TST44. INVALID ADDRESS TEST
FOR ALL ILLEGAL DSA ADDRESSES READ THE IAE BIT SET.
- TST45. ADDRESS OVERFLOW
TEST FOR AOE ON TRANSFERS WHICH EXTEND BEYOND THE LAST BLOCK.
- TST46. SYNC BUS PARITY
TEST FOR BAD PARITY DETECTION AND GOOD PARITY GENERATION.
- TST47. WRITE READ MEMORY ARRAY (M-BUS BLOCK MODE)
WRITE READ MEMORY VIA M-BUS BLOCK WITH MINIMUM OVERHEAD
- TST48. TEST THE CRC DATA BUS BETWEEN THE CRC GENERATORS AND THE CRC/MBUS DATA MUX FOR CONTINUITY AND BIT UNIQUENESS
- TST49. TEST CRC CODES GENERATED FOR ONE CRC GROUP (52 UNIQUE NIBBLES)
- TST50. TEST CRC CODES GENERATED FOR ONE

CRC GROUP (13 UNIBUS WORDS)

- TST51. TEST SYNDROME DECODE AND ERROR CORRECTION TO DECODE AND CORRECT SINGLE BIT AND MULTIPLE BIT CHANNEL ERRORS
- TST52. TEST SYNDROME DECODE TO DETECT BUT NOT CORRECT UNCORRECTABLE CHANNEL ERRORS
- TST53. TEST SYNDROME DECODE TO DETECT BUT NOT CORRECT UNCORRECTABLE MULTIPLE CHANNEL ERRORS
- TST54. TEST SYNDROME GENERATION, SYNDROME DECODE AND ERROR CORRECTION FOR SINGLE BIT CHANNEL ERRORS
- TST55. TEST SYNDROME GENERATION, SYNDROME DECODE AND ERROR CORRECTION FOR MULTIPLE BIT CHANNEL ERRORS
- TST56. TEST THE ECC ERROR REGISTER FOR CLEARING AND LATCHING OF ECC ERROR INFORMATION ON DETECTION OF ECC ERRORS
- TST57. TEST THE ECC ERROR LOCATION REGISTER FOR CLOCKING, BIT UNIQUENESS, CLEARING AND LATCHING
- TST58. VIA ECC_DM AND ECC_DIS TEST THE ECH BIT FOR SETTING AND NOT SETTING
- TST59. VIA ECC_EN, ECC_DM AND ECC_DIS TEST THE DCK_BIT FOR SETTING AND NOT SETTING
- TST60. PROM DATA TEST
VERIFY THAT CHECK SUM VALUES FOR ALL PROM LOCATIONS ARE CORRECT.

```

1          .SBTTL PROGRAM HEADER AND TABLES
33
35 000000          .ENABL ABS,AMA
36          002000          =          2000
38
39 002000          BGNMOD
40
41          :++
42          : THE PROGRAM HEADER IS THE INTERFACE BETWEEN
43          : THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
44          :--
45
46 002000          POINTER AL'.
47
64          HEADER ML11,A,0,1800.,0
65 002000
66
77          :
78          : NAMES OF DEVICES SUPPORTED BY THIS PROGRAM
79          :
80 002122          DEVTYP <ML-11>
81
82
83          :
84          :
85          : TEST DESCRIPTION
86          :
87 002130          DESCRIPT <ML-11 LOGIC TEST>
88
89
90          :
91          :
92          : THE GLOBAL ERROR TABLE (INFORMATION
93          : USED IN A CALL TO THE MACRO 'ERROR')
94          :
95
96 002152          ERRTBL
97          002152          000000          ERRNBR::          .WORD          0
98          002154          000000          ERRMSG::          .WORD          0
99          002156          000000          ERRBLK::          .WORD          0
100          002160          000000
101
102          :++
103          : THE DISPATCH TABLE CONTAINS THE STARTING ADDRES OF EACH TEST.
104          : IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
105          :--
106          :
107          :
108          : DISPATCH 60
109
110          :
111          :
112          :++
113          : THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
114          : THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
115          : IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
116          : AND IS USED AS A 'TEMPLATE' FOR BUILDING THE P-TABLES.
117          :--
118

```

```
119 002354          BGNHW  DFPTBL
120
130
131 002356 176400   .WORD 176400      :RH ADDRESS
132 002360 000070   .WORD 70          :RH TYPE
133 002362 000204   .WORD 204        :RH VECTOR ADDRESS
134 002364 000020   .WORD 16.        :NUMBER OF ARRAY MODULES
135 002366 000001   .WORD 1          :IS DRIVE OPTION ML11A, 1=16K, 0=64K
136 002370 000000   .WORD 0          :ML-11 DRIVE NUMBER
137 002372 000000   .WORD 0          :IS PARITY DISABLED, 1=YES, 0=NO
```

```
138
139 002374          ENDPHW
```

```
140
141
142
143 :++
144 : THE DEFAULT SOFTWARE P-TABLE CONTAINS VARIOUS DATA USED BY THE
145 : PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE SET
146 : UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR AT RUN
147 : TIME.
148 :--
```

```
149
150 002374          BGNSW  SFPTBL
151
159
160 002376 000000   PRSN: .WORD 0          :PRINT SERIAL NUMBER, 1=YES, 0=NO
161
162 002400          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

002400

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 MSGH4,6,0,77,1,16,,YES
GPRML MSGH5,10,1,YES
GPRMD MSGH6,12,0,7,0,7,YES
GPRML MSGH7,14,1,YES

ENDHRD

122	110	040	MSGH1:	.ASCIZ	/RH ADDRESS?/
111	123	040	MSGH2:	.ASCIZ	/IS RH AN '70' OR '11?/
122	110	040	MSGH3:	.ASCIZ	/RH VECTOR ADDRESS?/
116	125	115	MSGH4:	.ASCIZ	/NUMBER OF ARRAY MODULES?/
111	123	040	MSGH5:	.ASCIZ	/IS DRIVE OPTION AN ML11A?/
115	114	055	MSGH6:	.ASCIZ	/ML-11 DRIVE NUMBER?/
111	123	040	MSGH7:	.ASCIZ	/IS PARITY DISABLED?/
					.EVEN

:++
: 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.
:--

:++
: 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.??	;PRINT DRIVE SERIAL NUMBER?
					.EVEN	

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;
23
24
25
26
27
28
32 004152 000207      LRPT:  RTS      PC      ;      0716
33
34      ; Routine Size: 1 word
35      ; Maximum stack depth per invocation: 0 words
36
37
38
39
40
41
42
43
44
45
49 004154 004767 177772  L&RPT:: JSR      PC,LRPT      ;      0725
50 004160 104425
51 004162 000207      TRAP      25
52      RTS      PC
53
54      ; Routine Size: 4 words
55      ; Maximum stack depth per invocation: 0 words
56
57
58
59
60
61
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;
76
77
78
79
83 004164 000207      LAUTO:  RTS      PC      ;      0726
84
85      ; Routine Size: 1 word
86      ; Maximum stack depth per invocation: 0 words
87
88
89
90
91
92
93
94
95
96
100 004166 004767 177772  L&AUTO::JSR      PC,LAUTO      ;      0736

```


101 004172 104461
102 004174 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
118 : 0741 ! TO NO LONGER BE TESTED.
119 : 0742 !-
120 : 0743

121 : 0744 BGNDU;
122 : 0745 RETURN;
123 : 0746 ENDDU;

127

131 004176 000207

LDU: RTS PC ;

0737

132
133

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

134
139
140

144

148 004200 004767 177772

LSDU:: JSR PC,LDU ;

0745

149 004204 104453
150 004206 000207

TRAP 53
RTS PC

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

151
152
153
158
159

160 : 0747

161 : 0748 !+
162 : 0749 ! THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
163 : 0750 ! TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
164 : 0751 ! TO THE TEST CYCLE.
165 : 0752 !-
166 : ML3

167 : ML3
168 :
169 :

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

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

170 : 0753
171 : 0754 BGNAU;
172 : 0755 RETURN;
173 : 0756 ENDAU;

177

181 004210 000207

LAU: RTS PC ;

0746

182
183

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

184
189
190
194

198 004212 004767 177772
199 004216 104452
200 004220 000207
201
202
203
208
209
210 : 0757 END
211 : 0758
212 : 0759 ELUDOM
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230

LSAU:: JSR PC,LAU ;
TRAP 52
RTS PC

0755

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

:ML3
:

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

: Size: 20 code + 0 data words
: Run Time: 00:01.9
: Elapsed Time: 00:07.1
: Memory Used: 12 pages
: Compilation Complete

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (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 :
81 :
82 :
83 :
84 :
85 :
86 :
87 :
88 :
89 :
90 :
91 :
92 :
93 :

1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564

```
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, 4, 0],  
tes,  
DT1_LOAD =  
set  
EO_5 = [0, 0, 6, 0],  
CO_5 = [0, 6, 6, 0],  
CRC_DATA = [0, 13, 1, 0],  
SGL_DATA = [0, 14, 1, 0],  
UNC_DATA = [0, 15, 1, 0],  
EE_DATA = [0, 0, 16, 0],  
AO_5 = [1, 0, 6, 0],  
PO_5 = [1, 6, 6, 0],  
ECP_DATA = [1, 12, 1, 0],  
BO_5 = [2, 0, 6, 0],  
tes;
```

!<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (1)

!MAPS OWN STORAGE NIB_SAVE INTO TEN FOUR EIT 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>

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (2)

```
95 :ML4
96 :
97 :
98 : 1565 :
99 : 1566 : OWN STORAGE
100 : 1567 :
101 : 1568 :
102 : 1569 : own
103 : 1570 : NIB_SAVE : block [3] field (NIB_MAP) volatile,
104 : 1571 :
105 : 1572 : HW OR TBL : vector [127] volatile,
106 : 1573 : PTBL_PTR : volatile,
107 : 1574 : OP_NUM_ARR : volatile,
108 : 1575 : ARR_INC : volatile,
109 : 1576 : GOOD_BLK : volatile,
110 : 1577 : PAR_DIS : volatile,
111 : 1578 : CHIP_SIZ : volatile,
112 : 1579 : LST_BLK : volatile,
113 : 1580 : ARR_16 : volatile,
114 : 1581 : LST_ARR : volatile,
115 : 1582 : IO_BUF : vector [256] volatile,
116 : 1583 : STR_OFF : vector [10, byte] volatile,
117 : 1584 : stack : vector [198, byte] volatile,
118 : 1585 : PD_TEMP : bitvector [16] volatile,
119 : 1586 : W_T_SIZE : volatile,
120 : 1587 : RAS_INC : volatile,
121 : 1588 : WT_DATA : volatile,
122 : 1589 : RD_DATA : volatile,
123 : 1590 : DRIVE_TYPE : volatile,
124 : 1591 : LST_DOT_REG : volatile,
125 : 1592 : REG_INIT_FLG : initial (0) volatile,
126 : 1593 : A_CAL : volatile,
127 : 1594 : B_CAL : volatile,
128 : 1595 : P_CAL : volatile,
129 : 1596 : A_GEN : volatile,
130 : 1597 : B_GEN : volatile,
131 : 1598 : P_GEN : volatile,
132 : 1599 :
133 : 1600 : <BLF/NOFORMAT>
134 : 1601 :
135 : 1602 : ML_REG: blockvector [NUM_OF_REG, FIELD_SIZ] field(WORD_MAP) !ML11 REGISTERS
136 : 1603 : preset (
137 : 1604 :
138 : 1605 :
139 : 1606 :
140 : 1607 :
141 : 1608 :
142 : 1609 : [0, FORCE_HI] = %o'004000', !MLCS1
143 : 1610 :
144 : 1611 : [0, FORCE_LO] = %o'173701',
145 : 1612 : [0, DONT_CARE] = %o'160200',
146 : 1613 :
147 : 1614 : [5, FORCE_LO] = %o'25077', !MLDS
148 : 1615 : [5, FORCE_HI] = %o'010600',
149 : 1616 : [5, DONT_CARE] = %o'000100',
```

! STORAGE LOCATION TO SAVE NIBBLE DATA READ DURING DIAG MODE
! STORES HARDWARE ORED FROM 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 XFRS
! ROW ADRS STROBE INCREMENT
! SAVE WRITE DATA DURING REG READ WRITE TEST
! SAVE READ DATA DURING REG READ WRITE TEST
! DRIVE TYPE STORAGE LOCATION
! LAST ML-11 REG INDEX FOR THIS TYPE RH CONTROLLER
! FLAG TO DETECT DOING REG INIT TEST
! CALCULATED CRC STORAGE LOCATION
! CALCULATED CRC STORAGE LOCATION
! CALCULATED CRC STORAGE LOCATION
! GENERATED CRC STORAGE LOCATION
! GENERATED CRC STORAGE LOCATION
! GENERATED CRC STORAGE LOCATION

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (2)

```
151 :ML4
152 :
153 :
154 : 1617
155 : 1618 [6,FORCE_LO] = %o'014620', !MLER
156 : 1619
157 : 1620 [7,DONT_CARE] = %o'177400', !MLAS
158 : 1621
159 : 1622 [8,FORCE_LO] = %o'100000', !MLPA
160 : 1623
161 : 1624 [10,FORCE_LO] = %o'000020', !MLMR
162 : 1625 [10,DONT_CARE] = %o'177400',
163 : 1626
164 : 1627 [11,FORCE_HI] = %o'000110', !MLDT
165 : 1628 [11,FORCE_LO] = %o'177666',
166 : 1629 [11,DONT_CARE] = %o'000001',
167 : 1630
168 : 1631 [13,FORCE_LO] = %o'140300', !MLE1
169 : 1632
170 : 1633 [14,FORCE_LO] = %o'100300', !MLE2
171 : 1634
172 : 1635 [17,FORCE_LO] = %o'010000', !MLEE
173 : 1636 [21,DONT_CARE] = %o'000000', !MLCS2
174 : 1637 J volatile,
175 : 1638
176 : 1639
```

```
177 : 1640 REM_TBL:vector [63,byte] !REMAINDER TABLE
178 : 1641 preset ( !STRUCTURE TO STORE PRECALCULATED
179 : 1642 [0] = %b'000001', !CRC REMAINDER VALUES FOR CRC CODE
180 : 1643 [1] = %b'000010', !GENERATION
181 : 1644 [2] = %b'000100',
182 : 1645 [3] = %b'001000',
183 : 1646 [4] = %b'010000',
184 : 1647 [5] = %b'100000',
185 : 1648 [6] = %b'000011',
186 : 1649 [7] = %b'000110',
187 : 1650 [8] = %b'001100',
188 : 1651 [9] = %b'011000',
189 : 1652 [10] = %b'110000',
190 : 1653 [11] = %b'100011',
191 : 1654 [12] = %b'000101',
192 : 1655 [13] = %b'001010',
193 : 1656 [14] = %b'010100',
194 : 1657 [15] = %b'101000',
195 : 1658 [16] = %b'010011',
196 : 1659 [17] = %b'100110',
197 : 1660 [18] = %b'001111',
198 : 1661 [19] = %b'011110',
199 : 1662 [20] = %b'111100',
200 : 1663 [21] = %b'111011',
201 : 1664 [22] = %b'110101',
202 : 1665 [23] = %b'101001',
203 : 1666 [24] = %b'010001',
204 : 1667 [25] = %b'100010',
205 : 1668 [26] = %b'000111',
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (2)

```
207 :ML4
208 :
209 :
210 : 1669 [27] = %b'001110'
211 : 1670 [28] = %b'011100'
212 : 1671 [29] = %b'111000'
213 : 1672 [30] = %b'110011'
214 : 1673 [31] = %b'100101'
215 : 1674 [32] = %b'001001'
216 : 1675 [33] = %b'010010'
217 : 1676 [34] = %b'100100'
218 : 1677 [35] = %b'001011'
219 : 1678 [36] = %b'010110'
220 : 1679 [37] = %b'101100'
221 : 1680 [38] = %b'011011'
222 : 1681 [39] = %b'110110'
223 : 1682 [40] = %b'101111'
224 : 1683 [41] = %b'011101'
225 : 1684 [42] = %b'111010'
226 : 1685 [43] = %b'110111'
227 : 1686 [44] = %b'101101'
228 : 1687 [45] = %b'011001'
229 : 1688 [46] = %b'110010'
230 : 1689 [47] = %b'100111'
231 : 1690 [48] = %b'001101'
232 : 1691 [49] = %b'011010'
233 : 1692 [50] = %b'110100'
234 : 1693 [51] = %b'101011'
235 : 1694 [52] = %b'010101'
236 : 1695 [53] = %b'101010'
237 : 1696 [54] = %b'010111'
238 : 1697 [55] = %b'101110'
239 : 1698 [56] = %b'011111'
240 : 1699 [57] = %b'111110'
241 : 1700 [58] = %b'111111'
242 : 1701 [59] = %b'111101'
243 : 1702 [60] = %b'111001'
244 : 1703 [61] = %b'110001'
245 : 1704 [62] = %b'100001' volatile,
246 : 1705
247 : 1706 DT_1:blockvector[5,3] field (DT1_FLD)
248 : 1707 preset (
249 : 1708 [0,E0_5] = %b'111111'
250 : 1709 [0,C0_5] = %b'100100'
251 : 1710 [0,CRC_DATA] = %b'1'
252 : 1711 [0,SGL_DATA] = %b'1'
253 : 1712 [0,UNC_DATA] = %b'0'
254 : 1713 [0,A0_5] = %b'000000'
255 : 1714 [0,P0_5] = %b'111111'
256 : 1715 [0,ECH_DATA] = %b'0'
257 : 1716 [0,B0_5] = %b'000000'
258 : 1717 [1,E0_5] = %b'111111'
259 : 1718 [1,C0_5] = %b'100100'
260 : 1719 [1,CRC_DATA] = %b'1'
261 : 1720 [1,SGL_DATA] = %b'1'
```

```
!DATA TABLE 1
!STRUCTURE TO STORE TEST DATA
!FOR TEST 56 'ECC ERROR LOCATION TEST'
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (2)

```
263 :ML4
264 :
265 :
266 : 1721 [1,UNC_DATA] = %b'0'
267 : 1722 [1,A0_5] = %b'111111'
268 : 1723 [1,PO_5] = %b'000000'
269 : 1724 [1,ECH_DATA] = %b'0'
270 : 1725 [1,BO_5] = %b'000000'
271 : 1726 [2,EO_5] = %b'111111'
272 : 1727 [2,CO_5] = %b'000101'
273 : 1728 [2,CRF_DATA] = %b'0'
274 : 1729 [2,SGL_DATA] = %b'0'
275 : 1730 [2,UNC_DATA] = %b'1'
276 : 1731 [2,A0_5] = %b'000000'
277 : 1732 [2,PO_5] = %b'111111'
278 : 1733 [2,ECH_DATA] = %b'1'
279 : 1734 [2,BO_5] = %b'111111'
280 : 1735 [3,EO_5] = %b'111111'
281 : 1736 [3,CO_5] = %b'000101'
282 : 1737 [3,CRF_DATA] = %b'0'
283 : 1738 [3,SGL_DATA] = %b'0'
284 : 1739 [3,UNC_DATA] = %b'1'
285 : 1740 [3,A0_5] = %b'000000'
286 : 1741 [3,PO_5] = %b'000000'
287 : 1742 [3,ECH_DATA] = %b'0'
288 : 1743 [3,BO_5] = %b'111111'
289 : 1744 [4,EO_5] = %b'111111'
290 : 1745 [4,CO_5] = %b'000101'
291 : 1746 [4,CRF_DATA] = %b'0'
292 : 1747 [4,SGL_DATA] = %b'0'
293 : 1748 [4,UNC_DATA] = %b'1'
294 : 1749 [4,A0_5] = %b'111111'
295 : 1750 [4,PO_5] = %b'000000'
296 : 1751 [4,ECH_DATA] = %b'1'
297 : 1752 [4,BO_5] = %b'111111')volatile,
298 : 1753
299 : 1754
300 : 1755 !<BLF/FORMAT>
301 : 1756
302 : 1757 RH_ADD : volatile,
303 : 1758 RH_TYP : volatile,
304 : 1759 RH_VEC : volatile,
305 : 1760 ML_LUN : volatile,
306 : 1761 ML_DUT : volatile;
307 : 1762
308 : 1763
309 : 1764 !<BLF/PAGE>
```

```
!RH CONTROLLER BASE ADDRESS
!RH CONTROLLER TYPE
!RH CONTROLLER VECTOR ADDRESS
!ML LOGICAL UNIT NO.
!ML DRIVE NUMBER
```


22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:(NEALE)BL3ML4.BLI.2 (3)

```

311 :ML4
312 :
313 :
314 : 1765 !
315 : 1766 EQUALS:
316 : 1767 !
317 : 1768 ! MACRO DEFINITIONS
318 : 1769 !
319 : 1770 !
320 : 1771 macro
321 : 1772 !
322 : 1773 ! REGISTER NAMES:
323 : 1774 !
324 : M 1775 .ML_CS1 =
325 : 1776 .ML_REG [0,REGISTER_ADD]%, !CONTROL AND STATUS REGISTER 1
326 : M 1777 .MLWC = !WORD COUNT REGISTER
327 : 1778 .ML_REG [1,REGISTER_ADD]%,
328 : M 1779 .MLBA = !UNIBUS ADDRESS REGISTER
329 : 1780 .ML_REG [2,REGISTER_ADD]%,
330 : M 1781 .MLDA = !DESIRED ADDRESS REGISTER
331 : 1782 .ML_REG [3,REGISTER_ADD]%,
332 : M 1783 .ML_CS2 =
333 : 1784 .ML_REG [4,REGISTER_ADD]%, !CONTROL AND STATUS REGISTER 2
334 : M 1785 .MLDS =
335 : 1786 .ML_REG [5,REGISTER_ADD]%, !DRIVE STATUS REGISTER
336 : M 1787 .MLER =
337 : 1788 .ML_REG [6,REGISTER_ADD]%, !ERROR REGISTER
338 : M 1789 .MLAS =
339 : 1790 .ML_REG [7,REGISTER_ADD]%, !ATTENTION SUMMARY REGISTER
340 : M 1791 .MLLA =
341 : 1792 .ML_REG [8,REGISTER_ADD]%, !LOOK AHEAD REGISTER
342 : M 1793 .MLPA =
343 : 1794 .ML_REG [8,REGISTER_ADD]%, !PROM ADDRESS REGISTER
344 : M 1795 .MLDB =
345 : 1796 .ML_REG [9,REGISTER_ADD]%, !DATA BUFFER REGISTER
346 : M 1797 .MLMR =
347 : 1798 .ML_REG [10,REGISTER_ADD]%, !MAINTENANCE REGISTER
348 : M 1799 .MLDT =
349 : 1800 .ML_REG [11,REGISTER_ADD]%, !DRIVE TYPE REGISTER
350 : M 1801 .MLSN =
351 : 1802 .ML_REG [12,REGISTER_ADD]%, !SERIAL NUMBER REGISTER
352 : M 1803 .MLE1 =
353 : 1804 .ML_REG [13,REGISTER_ADD]%, !ECC CRC WORD REGISTER 1
354 : M 1805 .MLE2 =
355 : 1806 .ML_REG [14,REGISTER_ADD]%, !ECC CRC WORD REGISTER 2
356 : M 1807 .MLD1 =
357 : 1808 .ML_REG [15,REGISTER_ADD]%, !DATA DIAGNOSTIC REGISTER 1
358 : M 1809 .MLD2 =
359 : 1810 .ML_REG [16,REGISTER_ADD]%, !DATA DIAGNOSTIC REGISTER 2
360 : M 1811 .MLEE =
361 : 1812 .ML_REG [17,REGISTER_ADD]%, !ECC ERROR REGISTER
362 : M 1813 .MLEL =
363 : 1814 .ML_REG [18,REGISTER_ADD]%, !ECC ERROR LOCATION REGISTER
364 : M 1815 .MLPD =
365 : 1816 .ML_REG [19,REGISTER_ADD]%, !PROM DATA REGISTER

```

```
367 ;ML4
368 ;
369 ;
370 ; M 1817 MLBAE =
371 ; 1818 .ML_REG [20,REGISTER_ADD]%,
372 ; M 1819 MLCS3 =
373 ; 1820 .ML_REG [21,REGISTER_ADD]%,
374 ; 1821 ;
375 ; 1822 BIT ASSIGNMENTS:
376 ; 1823 ;
377 ; M 1824 SC =
378 ; 1825 (MLCS1)<15,1>%
379 ; M 1826 TRE =
380 ; 1827 (MLCS1)<14,1>%
381 ; M 1828 MCPE =
382 ; 1829 (MLCS1)<13,1>%
383 ; M 1830 DVA =
384 ; 1831 (MLCS1)<11,1>%
385 ; M 1832 RDY =
386 ; 1833 (MLCS1)<7,1>%
387 ; M 1834 IE =
388 ; 1835 (MLCS1)<6,1>%
389 ; M 1836 GO =
390 ; 1837 (MLCS1)<0,1>%
391 ; M 1838 ML_FUNC =
392 ; 1839 (MLCS1)<0,6>%
393 ; M 1840 DLT =
394 ; 1841 (MLCS2)<15,1>%
395 ; M 1842 WCE =
396 ; 1843 (MLCS2)<14,1>%
397 ; M 1844 PE =
398 ; 1845 (MLCS2)<13,1>%
399 ; M 1846 NED =
400 ; 1847 (MLCS2)<12,1>%
401 ; M 1848 NEM =
402 ; 1849 (MLCS2)<11,1>%
403 ; M 1850 PGE =
404 ; 1851 (MLCS2)<10,1>%
405 ; M 1852 MXF =
406 ; 1853 (MLCS2)<9,1>%
407 ; M 1854 MDPE =
408 ; 1855 (MLCS2)<8,1>%
409 ; M 1856 ORDY =
410 ; 1857 (MLCS2)<7,1>%
411 ; M 1858 IRDY =
412 ; 1859 (MLCS2)<6,1>%
413 ; M 1860 CLR =
414 ; 1861 (MLCS2)<5,1>%
415 ; M 1862 PAT =
416 ; 1863 (MLCS2)<4,1>%
417 ; M 1864 BAI =
418 ; 1865 (MLCS2)<3,1>%
419 ; M 1866 DRV_NUM =
420 ; 1867 (MLCS2)<0,3>%
421 ; M 1868 ATTN =
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (3)

!BUS ADDRESS EXTENSION REGISTER

!CONTROL AND STATUS REGISTER 3

!MLCS1 BIT ASSIGNMENTS

!MLCS2 BIT ASSIGNMENTS

423 :ML4

424 :

425 :

426 :

427 :

428 :

429 :

430 :

431 :

432 :

433 :

434 :

435 :

436 :

437 :

438 :

439 :

440 :

441 :

442 :

443 :

444 :

445 :

446 :

447 :

448 :

449 :

450 :

451 :

452 :

453 :

454 :

455 :

456 :

457 :

458 :

459 :

460 :

461 :

462 :

463 :

464 :

465 :

466 :

467 :

468 :

469 :

470 :

471 :

472 :

473 :

474 :

475 :

476 :

477 :

M 1869 (MLDS)<15,1>%,
M 1870 COMP_ERR =
M 1871 (MLDS)<14,1>%,
M 1872 MOL =
M 1873 (MLDS)<12,1>%,
M 1874 LBT =
M 1875 (MLDS)<10,1>%,
M 1876 DPR =
M 1877 (MLDS)<8,1>%,
M 1878 DRY =
M 1879 (MLDS)<7,1>%,
M 1880 VV =
M 1881 (MLDS)<6,1>%,
M 1882 DCK =
M 1883 (MLER)<15,1>%,
M 1884 UNS =
M 1885 (MLER)<14,1>%,
M 1886 OPI =
M 1887 (MLER)<13,1>%,
M 1888 IAE =
M 1889 (MLER)<10,1>%,
M 1890 AOE =
M 1891 (MLER)<9,1>%,
M 1892 ECH_ERR =
M 1893 (MLER)<6,1>%,
M 1894 DPAR =
M 1895 (MLER)<5,1>%,
M 1896 CPAR =
M 1897 (MLER)<3,1>%,
M 1898 RPR =
M 1899 (MLER)<2,1>%,
M 1900 ILR =
M 1901 (MLER)<1,1>%,
M 1902 ILF =
M 1903 (MLER)<0,1>%,
M 1904 ARR_TYP =
M 1905 (MLMR)<10,1>%,
M 1906 ML_NUM_ARR =
M 1907 (MLMR)<11,5>%,
M 1908 REF_MAR =
M 1909 (MLMR)<7,1>%,
M 1910 PROM_RW =
M 1911 (MLMR)<6,1>%,
M 1912 PROM_DIS =
M 1913 (MLMR)<5,1>%,
M 1914 DAT_CLK =
M 1915 (MLMR)<4,1>%,
M 1916 DAT_DM =
M 1917 (MLMR)<3,1>%,
M 1918 DCK_EN =
M 1919 (MLMR)<2,1>%,
M 1920 ECC_DIS =

22-Dec-1980 09:24:31

22-Dec-1980 09:21:22

TOPC-20 Bliss-16 V2(212)

PA:<NEALE>BL3ML4.BLI.2 (3)

!MLDS BIT ASSIGNMENTS

!MLER BIT ASSIGNMENTS

!MLMR BIT ASSIGNMENTS

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (3)

```
479 :ML4
480 :
481 :
482 : 1921 (MLMR)<1,1>%,
483 : M 1922 ECC_DM =
484 : 1923 (MLMR)<0,1>%,
485 : M 1924 DRV_TYP =
486 : 1925 (MLDT)<0,1>%,
487 : M 1926 CRC_A =
488 : M 1927
489 : 1928 (MLE1)<0,6>%,
490 : M 1929 PAR_CRC_WRD =
491 : 1930 (MLE1)<8,65>%,
492 : M 1931 CRC_B =
493 : M 1932
494 : 1933 (MLE2)<0,6>%,
495 : M 1934 UNC_ERR =
496 : 1935 (MLEE)<15,1>%,
497 : M 1936 SGL_ERR =
498 : 1937 (MLEE)<14,1>%,
499 : M 1938 CRC_ERR =
500 : 1939 (MLEE)<13,1>%,
501 : M 1940 BIT_IN_ERR =
502 : 1941 (MLEE)<20,8>%,
503 : M 1942 CHAN_IN_ERR =
504 : 1943 (MLEE)<8,65>%,
505 : 1944
506 : 1945 MISCELLANEOUS MACRO DEFINITIONS:
507 : 1946
508 : M 1947 SN3 =
509 : 1948 (MLSN)<12,4>%,
510 : M 1949 SN2 =
511 : 1950 (MLSN)<8,4>%,
512 : M 1951 SN1 =
513 : 1952 (MLSN)<4,4>%,
514 : M 1953 SNO =
515 : 1954 (MLSN)<0,4>%,
516 : M 1955 IS_SET =
517 : M 1956
518 : 1957 eql 1%,
519 : M 1958 IS_NOT_SET =
520 : M 1959
521 : 1960 eql 0%,
522 : M 1961 HI =
523 : M 1962
524 : 1963 ML_REG[index,FORCE_HI]%,
525 : M 1964 LO =
526 : M 1965
527 : 1966 ML_REG[index,FORCE_LO]%,
528 : M 1967 IGNORE =
529 : M 1968
530 : 1969 ML_REG[index,DONT_CARE]%,
531 : M 1970 MLE2_MASK =
532 : M 1971
533 : 1972 ML_REG[14,DONT_CARE]%,
```

!MLDT BIT ASSIGNMENTS

!MLE1 BIT ASSIGNMENTS

!MLE2 BIT ASSIGNMENTS

!MLEE BIT ASSIGNMENTS

!TEST IF BIT IS EQUAL 1

!TEST IF BIT IS EQUAL 0

!READS REGISTERS FORCED HI BITS FROM PERSONALITY TABLE

!READS REGISTERS FORCED LO BITS FROM PERSONALITY TABLE

!READS REGISTERS DONT_CARE BITS FROM PERSONALITY TABLE

!READS MLE2 DONT CARE MASK EITHER DATA DIAG OR ECC CIE REG

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (3)

```

535 :ML4
536 :
537 :
538 : M 1973      WRT_MASK =
539 : M 1974      !GENERATE MASK DATA PATTERN USING REGISTER FORCE LO, HI AND IGNORE B
540 : M 1975      .IGNORE or ((not .LO) and (.HI or .TST_PAT))%,
541 : M 1976      CLR_MBUS =
542 : M 1977      !CLEAR MASS BUS RESTORE DRIVE NUMBER
543 : M 1978      CLR = ONE; DRV_NUM = .ML_DUT%,
544 : M 1979      !<BLF/SYNONYM IS_SET = EQL 1 * >
545 : M 1980      !<BLF/SYNONYM IS_NOT_SET = EQL 0 * >
546 : M 1981
547 : M 1982      !DIAGNOSTIC DATA REGISTER MACROS
548 : M 1983
549 : M 1984      RD_LNG_WRD =
550 : M 1985      !READ DATA DIAG REGS INTO BIND LOCATIONS
551 : M 1986      D1_TEMP = .MLD1;
552 : M 1987      D2_TEMP = .MLD2;
553 : M 1988      E2_TEMP = .MLE2%,
554 : M 1989      WRT_LNG_WRD =
555 : M 1990      !LOADS DATA DIAG REG WITH CONTENTS OF BIND LOCATIONS
556 : M 1991      MLD1 = .D1_TEMP;
557 : M 1992      MLD2 = .D2_TEMP;
558 : M 1993      MLE2 = .E2_TEMP%,
559 : M 1994      TIME_OUT_LOOP =
560 : M 1995      !WAIT LOOP DURING MASS BUS TRANSFER
561 : M 1996      do
562 : M 1997      0
563 : M 1998      until .DRY;%
564 : M 1999
565 : M 2000      !
566 : M 2001      ! BIND DECLARATIONS
567 : M 2002      !
568 : M 2003
569 : M 2004      bind
570 : M 2005      !
571 : M 2006      ! ERROR DATA MAPPING FORMATS
572 : M 2007      !
573 : M 2008      FMT_1 = uplit (%asciz'%AEXPECTED: %06% READ: %06%N%N'),
574 : M 2009      FMT_2 = uplit (%asciz'%AGOOD DATA: %06% BAD DATA: %06% XOR: %06%N%N'),
575 : M 2010      FMT_3 = uplit (%asciz'%ADRIVE SN: %06%N%N'),
576 : M 2011      FMT_4 = uplit (%asciz'%ABIT IN ERROR: %06%N%N'),
577 : M 2012      FMT_5 = uplit (%asciz'%AGOOD NIB DATA: %02% BAD NIB DATA: %02% NIB POS: %04%N%N'),
578 : M 2013      FMT_6 = uplit (%asciz'%ANIB IN ERROR: %04%N%N'),
579 : M 2014      FMT_7 = uplit (%asciz'%AFAILED AT: %06%N%N'),
580 : M 2015      FMT_8 = uplit (%asciz'%AREPLACE ARR MOD: %02%N%N'),
581 : M 2016      FMT_9 = uplit (%asciz'%AFAILED AT DSA: %06%N%N'),
582 : M 2017      FMT_10 = uplit (%asciz'%ABIT<15:10>: %06% BIT<9:0>: %06%N%N'),
583 : M 2018      FMT_11 = uplit (%asciz'%AFAILING REG ADRS: %06%N%N'),
584 : M 2019      FMT_12 = uplit (%asciz'%AFAILING FUNC: %06%N%N'),
585 : M 2020      FMT_13 = uplit (%asciz'%AOFF SET CNT FOR NIB : %02% = %02%N%N'),
586 : M 2021      FMT_14 = uplit (%asciz'%AWRITE: %02% READ: %02%N%N'),
587 : M 2022      FMT_15 = uplit (%asciz'%ANIBBLES XFERED BEFORE ERROR: %03%N'),
588 : M 2023      FMT_16 = uplit (%asciz'%AFAILING REG: %06% GOOD DATA: %06% BAD DATA: %06%N%N'),
589 : M 2024      FMT_17 = uplit (%asciz'%ADIAGNOSING UNIT %01%N%N'),

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 .2(212)
PA: <NEALE>BL3ML4.BLI.2 (3)

591 :ML4
592 :
593 :
594 :
595 :
596 :
597 :
598 :
599 :
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 :

2025 FMT_18 = uplit (%asciz'XATIMED OUT DURING MBUS %02XA FUNC%N%N'),
2026 FMT_19 = uplit (%asciz'XACRC GEN = B: %B6XA A: %B6XA P: %B6%N'),
2027 FMT_20 = uplit (%asciz'XACRC CAL = B: %B6XA A: %B6XA P: %B6%N%N'),
2028 FMT_21 = uplit (%asciz'XAF AILED AT PLOG: %D2XA CHANNEL: %D2%N%N'),
2029 FMT_22 = uplit (%asciz'XAF AILED AT WRD: %D2XA BIT: %D2%N%N'),
2030 FMT_23 = uplit (%asciz'XAREGISTER%3XAADDRESS%5XACONTENTS%N%N'),
2031 FMT_24 = uplit (%asciz'XS%T%5%06XS4%016%N'),
2032 FMT_25 = uplit (%asciz'XS4%T%A:XS%06'),
2033 FMT_26 = uplit (%asciz'XS4%T%A:XS%D1%01%D1%D1%N'),

...
... ERROR MESSAGE MAPPING FORMATS

2034 :
2035 :
2036 :
2037 ONE_FMT = uplit (%asciz'XT%N'),
2038 TWO_FMT = uplit (%asciz'XTXT%N'),
2039 THR_FMT = uplit (%asciz'XTXTXT%N'),
2040 FOR_FMT = uplit (%asciz'XTXTXTXT%N'),
2041 FIV_FMT = uplit (%asciz'XTXTXTXTXT%N'),
2042 SIX_FMT = uplit (%asciz'XTXTXTXTXTXT%N'),
2043 SEV_FMT = uplit (%asciz'XTXTXTXTXTXTXT%N'),
2044 EIG_FMT = uplit (%asciz'XTXTXTXTXTXTXTXT%N'),
2045 NIN_FMT = uplit (%asciz'XTXTXTXTXTXTXTXTXT%N'),
2046 TEN_FMT = uplit (%asciz'XTXTXTXTXTXTXTXTXTXT%N'),
2047 ELV_FMT = uplit (%asciz'XTXTXTXTXTXTXTXTXTXTXT%N'),

...
... DIAGNOSTIC VOCABULARY

...
... WORDS

2051 :
2052 :
2053 :
2054 WRD_1 = uplit (%asciz' GO'),
2055 WRD_2 = uplit (%asciz' DRV RDY'),
2056 WRD_3 = uplit (%asciz' ILF'),
2057 WRD_4 = uplit (%asciz' OPI'),
2058 WRD_5 = uplit (%asciz' BAD'),
2059 WRD_6 = uplit (%asciz' GOOD'),
2060 WRD_7 = uplit (%asciz' PARITY NOT'),
2061 WRD_8 = uplit (%asciz' GENERATED'),
2062 WRD_9 = uplit (%asciz' DETECTED'),
2063 WRD_10 = uplit (%asciz' ERROR'),
2064 WRD_11 = uplit (%asciz' AFTER'),
2065 WRD_12 = uplit (%asciz' DURING'),
2066 WRD_13 = uplit (%asciz' AT'),
2067 WRD_14 = uplit (%asciz' FAILURE'),
2068 WRD_15 = uplit (%asciz' ATA'),
2069 WRD_16 = uplit (%asciz' ATTN'),
2070 WRD_17 = uplit (%asciz' WRITING'),
2071 WRD_18 = uplit (%asciz' VV'),
2072 WRD_19 = uplit (%asciz' FUNC'),
2073 WRD_20 = uplit (%asciz' TRE'),
2074 WRD_21 = uplit (%asciz' RMR'),
2075 WRD_22 = uplit (%asciz' EXCESSIVE'),
2076 WRD_23 = uplit (%asciz' MBUS'),

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (3)

647 :ML4
648 :
649 :
650 : 2077 WRD_24 = split (%asciz' DATA'),
651 : 2078 WRD_25 = split (%asciz' CONTINUITY'),
652 : 2079 WRD_26 = split (%asciz' AOE'),
653 : 2080 WRD_27 = split (%asciz' LBT'),
654 : 2081 WRD_29 = split (%asciz' PREMATURELY'),
655 : 2082 WRD_30 = split (%asciz' IAE'),
656 : 2083 WRD_31 = split (%asciz' INCREMENT'),
657 : 2084 WRD_32 = split (%asciz' WITH'),
658 : 2085 WRD_33 = split (%asciz' UV'),
659 : 2086 WRD_34 = split (%asciz' UNS'),
660 : 2087 WRD_35 = split (%asciz' PROM'),
661 : 2088 WRD_36 = split (%asciz' OR'),
662 : 2089 WRD_37 = split (%asciz' SELECT'),
663 : 2090 WRD_38 = split (%asciz' REG'),
664 : 2091 WRD_39 = split (%asciz' UNIQUE'),
665 : 2092 WRD_40 = split (%asciz' 14'),
666 : 2093 WRD_41 = split (%asciz' NIBBLE CNT'),
667 : 2094 WRD_42 = split (%asciz' GTR'),
668 : 2095 WRD_43 = split (%asciz' WHILE'),
669 : 2096 WRD_44 = split (%asciz' TRE'),
670 : 2097 WRD_45 = split (%asciz' INITIAL'),
671 : 2098 WRD_46 = split (%asciz' OFF SET'),
672 : 2099 WRD_47 = split (%asciz' COUNT'),
673 : 2100 WRD_48 = split (%asciz' DELAY'),
674 : 2101 WRD_49 = split (%asciz' TESTS'),
675 : 2102 WRD_50 = split (%asciz' ADRS'),
676 : 2103 WRD_51 = split (%asciz' COUNTER'),
677 : 2104 WRD_52 = split (%asciz' REG'),
678 : 2105 WRD_53 = split (%asciz' TESTED'),
679 : 2106 WRD_54 = split (%asciz' NIBBLE'),
680 : 2107 WRD_55 = split (%asciz' ALL'),
681 : 2108 WRD_56 = split (%asciz' TEST'),
682 : 2109 WRD_57 = split (%asciz' XFERED'),
683 : 2110 WRD_58 = split (%asciz' NIBBLES'),
684 : 2111 WRD_59 = split (%asciz' SC'),
685 : 2112 WRD_60 = split (%asciz' MULTIPLEXER'),
686 : 2113 WRD_61 = split (%asciz' UNEXPECTED'),
687 : 2114 WRD_62 = split (%asciz' NED'),
688 : 2115 WRD_63 = split (%asciz' ILR'),
689 : 2116 WRD_64 = split (%asciz' CRC'),
690 : 2117 WRD_65 = split (%asciz' SGL'),
691 : 2118 WRD_67 = split (%asciz' ECH'),
692 : 2119 WRD_68 = split (%asciz' UNC'),
693 : 2120 WRD_69 = split (%asciz' BIT'),
694 : 2121 WRD_70 = split (%asciz' CHANNEL'),
695 : 2122 WRD_71 = split (%asciz' LATCH'),
696 : 2123 WRD_72 = split (%asciz' DCK'),
697 : 2124 WRD_73 = split (%asciz' BUS'),
698 : 2125 WRD_74 = split (%asciz' SYNDROME'),
699 : 2126 WRD_75 = split (%asciz' DECODE'),
700 : 2127 WRD_76 = split (%asciz' NOT'),
701 : 2128 WRD_77 = split (%asciz' UNCORRECTABLE').

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bl iss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (3)

703 :ML4
704 :
705 :
706 :
707 :
708 :
709 :
710 :
711 :
712 :
713 :
714 :
715 :
716 :
717 :
718 :
719 :
720 :
721 :
722 :
723 :
724 :
725 :
726 :
727 :
728 :
729 :
730 :
731 :
732 :
733 :
734 :
735 :
736 :
737 :
738 :
739 :
740 :
741 :
742 :
743 :
744 :
745 :
746 :
747 :
748 :
749 :
750 :
751 :
752 :
753 :
754 :
755 :
756 :
757 :

2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180

```

: 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'),
PHR_11 = uplit (%asciz' NOT LATCHED'),
PHR_12 = uplit (%asciz' SINGLE BIT ERROR'),
PHR_13 = uplit (%asciz' MULTIPLE BIT ERROR'),
PHR_14 = uplit (%asciz' REGISTER DUMP'),
PHR_15 = uplit (%asciz' SERIAL #'),

: 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' MLPR'),
REG_5 = uplit (%asciz' MLAS'),
REG_6 = uplit (%asciz' MLDA'),
    
```


22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bl iss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (3)

```
759 :ML4
760 :
761 :
762 : 2181 REG_7 = uplit (%asciz' MLDT'),
763 : 2182 REG_8 = uplit (%asciz' MLPA'),
764 : 2183 REG_9 = uplit (%asciz' MLSN'),
765 : 2184 REG_10 = uplit (%asciz' MLE1'),
766 : 2185 REG_11 = uplit (%asciz' MLE2'),
767 : 2186 REG_12 = uplit (%asciz' MLD1'),
768 : 2187 REG_13 = uplit (%asciz' MLD2'),
769 : 2188 REG_14 = uplit (%asciz' MLEE'),
770 : 2189 REG_15 = uplit (%asciz' MLEL'),
771 : 2190 REG_16 = uplit (%asciz' MLPD'),
772 : 2191 REG_17 = uplit (%asciz' MLCS2'),
773 : 2192 REG_18 = uplit (%asciz' MLWC'),
774 : 2193 REG_19 = uplit (%asciz' MLBA'),
775 : 2194 REG_20 = uplit (%asciz' MLBAI'),
776 : 2195 REG_21 = uplit (%asciz' MLCS3'),
777 :
778 : 2196
779 : 2197 : MODULES IN ERROR MESSAGES
780 : 2198
781 : 2199 ASYNC = uplit (%asciz'ASYNCHRONOUS FAILURE MODULE 7361'),
782 : 2200 SYNC = uplit (%asciz'SYNCHRONOUS FAILURE MODULE 7362 '),
783 : 2201 ARR_DAT = uplit (%asciz'ARRAY DATA FAILURE MODULE 7363 '),
784 : 2202 MEM_ARR = uplit (%asciz'MEMORY ARRAY FAILURE MODULE 7357'),
785 : 2203 INTER = uplit (%asciz'INTERMEDIATE DIAGNOSTIC MESSAGE'),
786 : 2204 DATA_LATE = uplit (%asciz'DATA LATE ERROR DURING TRANSFER'),
787 : 2205 SC_SET = uplit (%asciz'SC BIT SET DURING TRANSFER'),
788 : 2206 TROUBLE_LOOP = uplit (%asciz'TROUBLE SHOOT LOOP ERRORS'),
789 : 2207 RH_ERROR = uplit (%asciz'RH CONTROLLER ERRORS'),
790 : 2208 TIME_OUT = uplit (%asciz'DRIVE HUNG AFTER MASS BUS TRANSFER'),
791 : 2209
792 : 2210 : DATA DIAGNOSTIC REGISTER SAVE LOCATIONS
793 : 2211
794 : 2212 D1_TEMP = NIB_SAVE,
795 : 2213 D2_TEMP = NIB_SAVE [1, 0, 16, 0],
796 : 2214 E2_TEMP = NIB_SAVE [2, 0, 16, 0];
797 : 2215
798 : 2216 !<BLF/PAGE>
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bl iss-16 V2(212)
PA:<NEALE>3L3ML4.BLI.2 (5)

```
799 :ML4
800 :
801 :
802 : 2217 routine LOAD_STACK (STK_PTR, NIB_PTR) : novalue =
803 : 2218   begin
804 : 2219
805 : 2220   ++
806 : 2221   FUNCTIONAL DESCRIPTION:
807 : 2222   LOAD STACK TAKES GOOD NIBBLE DATA
808 : 2223   FOUND IN THE STRUCTURE 'NIB_SAVE'
809 : 2224   AND STORES IT INTO THE STRUCTURE
810 : 2225   'STACK' REWRITING ANY BAD NIBBLE
811 : 2226   'STACK' LOCATIONS WITH GOOD NIBBLE
812 : 2227   DATA
813 : 2228
814 : 2229   FORMAL PARAMETERS:
815 : 2230   STK_PTR
816 : 2231   POINTS TO PRESENT DEPTH OF THE
817 : 2232   'STACK' WHERE PRESENT GOOD NIBBLE
818 : 2233   DATA IS TO BE STORED.
819 : 2234
820 : 2235   NIB_PTR
821 : 2236   POINTS TO CURRENT NIBBLE POSITION BEING
822 : 2237   MANIPULATED.
823 : 2238
824 : 2239   IMPLICIT INPUTS:
825 : 2240   STACK
826 : 2241   VECTOR OF 198 BYTE LOCATIONS WHERE
827 : 2242   GOOD NIBBLE DATA IS STORED
828 : 2243   DURING DIAGNOSTIC MODE READS, AFTER
829 : 2244   BAD NIBBLE LOCATIONS HAVE BEEN
830 : 2245   STRIPPED AWAY.
831 : 2246
832 : 2247   STK_OFF
833 : 2248
834 : 2249   vector of 9 byte LOCATIONS WHICH
835 : 2250   STORES AWAY A BAD NIBBLE OFF SET
836 : 2251   COUNT FOR EACH NIBBLE POSITION
837 : 2252
838 : 2253   NIB_SAVE
839 : 2254   BLOCK OF 3 WORDS TO STORE THE
840 : 2255   DATA FOUND IN MLD1, MLD2 AND
841 : 2256   MLE2 AFTER A DIAGNOSTIC MODE READ.
842 : 2257
843 : 2258   IMPLICIT OUTPUTS:
844 : 2259   'STACK' LOADED WITH GOOD NIBBLE
845 : 2260   DATA
846 : 2261
847 : 2262   COMPLETETION CODES:  NONE
848 : 2263
849 : 2264   SIDE EFFECTS:      NONE
850 : 2265
851 : 2266   --
852 : 2267
853 : 2268   case .NIB_PTR from 0 to 9 of
```

!SELECT NIBBLE DATA TO BE LOADED INTO THE STACK

22-Dec-1980 09:24:31 TOPS-20 Bl iss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (5)

855 ;ML4
856 :
857 :
858 :
859 :
860 :
861 :
862 :
863 :
864 :
865 :
866 :
867 :
868 :
869 :
870 :
871 :
872 :
873 :
874 :
875 :
876 :
877 :
878 :
879 :
880 :
881 :
882 :
883 :
884 :
885 :
886 :
887 :
888 :
889 :
890 :
891 :
892 :
893 :
894 :
895 :
896 :
897 :
898 :
899 :
900 :
901 :
905 :
906 :
907 :

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

```
set  
[0] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 0];  
      !LOAD NIBBLE DATA 0 INTO SELECTED STACK LOCATION  
[1] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 1];  
      !LOAD NIBBLE DATA 1 INTO SELECTED STACK LOCATION  
[2] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 2];  
      !LOAD NIBBLE DATA 2 INTO SELECTED STACK LOCATION  
[3] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 3];  
      !LOAD NIBBLE DATA 3 INTO SELECTED STACK LOCATION  
[4] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 4];  
      !LOAD NIBBLE DATA 4 INTO SELECTED STACK LOCATION  
[5] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 5];  
      !LOAD NIBBLE DATA 5 INTO SELECTED STACK LOCATION  
[6] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 6];  
      !LOAD NIBBLE DATA 6 INTO SELECTED STACK LOCATION  
[7] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 7];  
      !LOAD NIBBLE DATA 7 INTO SELECTED STACK LOCATION  
[8] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 8];  
      !LOAD NIBBLE DATA 8 INTO SELECTED STACK LOCATION  
[9] : stack [(.STK_PTR - (.STK_OFF [.NIB_PTR]))] = .NIB_SAVE [NIB 9];  
      !LOAD NIBBLE DATA 9 INTO SELECTED STACK LOCATION  
tes;  
end;
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

909          :ML4
910          :
911
912 004222   045   101   105   P.AAA:  .ASCII  /%AE/
913 004225   130   120   105     .ASCII  /XPE/
914 004230   103   124   105     .ASCII  /CTE/
915 004233   104   072   040     .ASCII  /D: /
916 004236   045   117   066     .ASCII  /%06/
917 004241   045   101   040     .ASCII  /%A /
918 004244   040   040   040     .ASCII  / /
919 004247   122   105   101     .ASCII  /REA/
920 004252   104   072   040     .ASCII  /D: /
921 004255   045   117   066     .ASCII  /%06/
922 004260   045   116   045     .ASCII  /%N%/
923 004263   116   000   000     .ASCII  /N/<00><00>
924 004266   045   101   107   P.AAB:  .ASCII  /%AG/
925 004271   117   117   104     .ASCII  /000/
926 004274   040   104   101     .ASCII  / DA/
927 004277   124   101   072     .ASCII  /TA:/
928 004302   040   045   117     .ASCII  / %0/
929 004305   066   045   101     .ASCII  /6%A/
930 004310   040   040   040     .ASCII  / /
931 004313   040   102   101     .ASCII  / BA/
932 004316   104   040   104     .ASCII  /D D/
933 004321   101   124   101     .ASCII  /ATA/
934 004324   072   040   045     .ASCII  /: %/
935 004327   117   066   045     .ASCII  /06%/
936 004332   101   040   040     .ASCII  /A /
937 004335   040   040   130     .ASCII  / X/
938 004340   117   122   072     .ASCII  /OR:/
939 004343   040   045   117     .ASCII  / %0/
940 004346   066   045   116     .ASCII  /6%N/
941 004351   045   116   000     .ASCII  /%N/<00>
942 004354   045   101   104   P.AAC:  .ASCII  /%AD/
943 004357   122   111   126     .ASCII  /RIV/
944 004362   105   040   123     .ASCII  /E S/
945 004365   116   072   040     .ASCII  /N: /
946 004370   045   117   066     .ASCII  /%06/
947 004373   045   116   045     .ASCII  /%N%/
948 004376   116   000   000     .ASCII  /N/<00>
949 004400   045   101   102   P.AAD:  .ASCII  /%AB/
950 004403   111   124   040     .ASCII  /IT /
951 004406   111   116   040     .ASCII  /IN /
952 004411   105   122   122     .ASCII  /ERR/
953 004414   117   122   072     .ASCII  /OR:/
954 004417   040   045   117     .ASCII  / %0/
955 004422   066   045   116     .ASCII  /6%N/
956 004425   045   116   000     .ASCII  /%N/<00>
957 004430   045   101   107   P.AAE:  .ASCII  /%AG/
958 004433   117   117   104     .ASCII  /000/
959 004436   040   116   111     .ASCII  / NI/
960 004441   102   040   104     .ASCII  /B D/
961 004444   101   124   101     .ASCII  /ATA/
962 004447   072   040   045     .ASCII  /: %/
963 004452   117   062   045     .ASCII  /02%/

```

```

965          :ML_4
966          :
967
968 004455    101    040    040    .ASCII /A /
969 004460    040    040    102    .ASCII / B/
970 004463    101    104    140    .ASCII /AD /
971 004466    116    111    102    .ASCII /NIB/
972 004471    040    104    101    .ASCII / DA/
973 004474    124    101    072    .ASCII /TA:/
974 004477    040    045    117    .ASCII / %0/
975 004502    062    045    101    .ASCII /2%A/
976 004505    040    040    040    .ASCII / /
977 004510    040    116    111    .ASCII / NI/
978 004513    102    040    120    .ASCII /B P/
979 004516    117    123    072    .ASCII /OS:/
980 004521    040    045    117    .ASCII / %0/
981 004524    064    045    116    .ASCII /4%N/
982 004527    045    116    000    .ASCII /%N/<00>
983 004532    045    101    116    P.AAF: .ASCII /%A/
984 004535    111    102    040    .ASCII /IB /
985 004540    111    116    040    .ASCII /IN /
986 004543    105    122    122    .ASCII /ERR/
987 004546    117    122    072    .ASCII /OR:/
988 004551    040    045    104    .ASCII / %0/
989 004554    064    045    116    .ASCII /4%N/
990 004557    045    116    000    .ASCII /%N/<00>
991 004562    045    101    106    P.AAG: .ASCII /%A/
992 004565    101    111    114    .ASCII /AIL/
993 004570    105    104    040    .ASCII /ED /
994 004573    101    124    072    .ASCII /AT:/
995 004576    040    045    117    .ASCII / %0/
996 004601    066    045    116    .ASCII /6%N/
997 004604    045    116    000    .ASCII /%N/<00>
998 004607    000    .ASCII <00>
999 004610    045    101    122    P.AAH: .ASCII /%A/
1000 004613    105    120    114    .ASCII /EPL/
1001 004616    101    103    105    .ASCII /ACE/
1002 004621    040    101    122    .ASCII / AR/
1003 004624    122    040    115    .ASCII /R M/
1004 004627    117    104    072    .ASCII /OD:/
1005 004632    040    045    104    .ASCII / %0/
1006 004635    062    045    116    .ASCII /2%N/
1007 004640    045    116    000    .ASCII /%N/<J0>
1008 004643    000    .ASCII <00>
1009 004644    045    101    106    P.AAI: .ASCII /%A/
1010 004647    101    111    114    .ASCII /AIL/
1011 004652    105    104    040    .ASCII /ED /
1012 004655    101    124    040    .ASCII /AT /
1013 004660    104    123    101    .ASCII /DSA/
1014 004663    072    040    045    .ASCII /: %/
1015 004666    117    066    045    .ASCII /06%/
1016 004671    116    045    116    .ASCII /N%N/
1017 004674    000    000    .ASCII <00><00>
1018 004676    045    101    102    P.AAJ: .ASCII /%A/
1019 004701    111    124    074    .ASCII /IT</

```

```

1021          ;ML4
1022          ;
1023
1024 004704    061    065    072    .ASCII /15:/
1025 004707    061    060    076    .ASCII /10>/
1026 004712    072    040    045    .ASCII /: %/
1027 004715    102    066    045    .ASCII /B6%/
1028 004720    101    040    040    .ASCII /A /
1029 004723    040    040    102    .ASCII / B/
1030 004726    111    124    074    .ASCII /IT</
1031 004731    071    072    060    .ASCII /9:0/
1032 004734    076    072    040    .ASCII />: /
1033 004737    045    102    061    .ASCII /%91/
1034 004742    060    045    116    .ASCII /0%N/
1035 004745    045    116    000    .ASCII /%N/<00>
1036 004750    045    101    106    P.AAK: .ASCII /%AF/
1037 004753    101    111    114    .ASCII /AIL/
1038 004756    111    116    107    .ASCII /ING/
1039 004761    040    122    105    .ASCII / RE/
1040 004764    107    040    101    .ASCII /G A/
1041 004767    104    122    123    .ASCII /DRS/
1042 004772    072    040    045    .ASCII /: %/
1043 004775    117    066    045    .ASCII /06%/
1044 005000    116    045    116    .ASCII /N%N/
1045 005003    000                .ASCII <00>
1046 005004    045    101    106    P.AAL: .ASCII /%AF/
1047 005007    101    111    114    .ASCII /AIL/
1048 005012    111    116    107    .ASCII /ING/
1049 005015    040    106    125    .ASCII / FU/
1050 005020    116    103    072    .ASCII /NC:/
1051 005023    040    045    117    .ASCII / %0/
1052 005026    066    045    116    .ASCII /6%N/
1053 005031    045    116    000    .ASCII /%N/<00>
1054 005034    040    045    101    P.AAM: .ASCII / %A/
1055 005037    117    106    106    .ASCII /OFF/
1056 005042    137    123    105    .ASCII / SE/
1057 005045    124    040    103    .ASCII /T C/
1058 005050    116    124    040    .ASCII /NT /
1059 005053    106    117    122    .ASCII /FOR/
1060 005056    040    116    111    .ASCII / NI/
1061 005061    102    040    072    .ASCII /B :/
1062 005064    040    045    104    .ASCII / %D/
1063 005067    062    040    045    .ASCII /2 %/
1064 005072    101    040    075    .ASCII /A =/
1065 005075    040    045    104    .ASCII / %D/
1066 005100    062    040    045    .ASCII /2 %/
1067 005103    116    045    116    .ASCII /N%N/
1068 005106    000    000                .ASCII <00><00>
1069 005110    045    101    127    P.AAN: .ASCII /%AW/
1070 005113    122    117    124    .ASCII /ROT/
1071 005116    105    072    040    .ASCII /E: /
1072 005121    045    104    062    .ASCII /%D2/
1073 005124    045    101    040    .ASCII /%A /
1074 005127    040    040    040    .ASCII / /
1075 005132    122    105    101    .ASCII /REA/

```

```

1077      :ML4
1078      :
1079
1080 005135 104 072 040 .ASCII /D: /
1081 005140 045 104 062 .ASCII /XD2/
1082 005143 045 116 045 .ASCII /XN%/
1083 005146 116 000 .ASCII /N/<00>
1084 005150 045 101 116 P.AAO: .ASCII /XAN/
1085 005153 111 102 102 .ASCII /IBB/
1086 005156 114 105 123 .ASCII /LES/
1087 005161 040 130 106 .ASCII / XF/
1088 005164 105 122 105 .ASCII /ERE/
1089 005167 104 040 102 .ASCII /D B/
1090 005172 105 106 117 .ASCII /EFO/
1091 005175 122 105 040 .ASCII /RE /
1092 005200 105 122 122 .ASCII /ERR/
1093 005203 117 122 072 .ASCII /OR:/
1094 005206 040 045 104 .ASCII / XD/
1095 005211 063 045 116 .ASCII /3XN/
1096 005214 000 000 .ASCII <00><00>
1097 005216 045 101 106 P.AAP: .ASCII /XAF/
1098 005221 101 111 114 .ASCII /AII/
1099 005224 111 116 107 .ASCII /ING/
1100 005227 040 122 105 .ASCII / RE/
1101 005232 107 072 040 .ASCII /G: /
1102 005235 045 117 066 .ASCII /X06/
1103 005240 045 101 040 .ASCII /XA /
1104 005243 107 117 117 .ASCII /GOO/
1105 005246 104 040 104 .ASCII /D D/
1106 005251 101 124 101 .ASCII /ATA/
1107 005254 072 040 045 .ASCII /: %/
1108 005257 117 066 045 .ASCII /O6%/
1109 005262 101 040 102 .ASCII /A B/
1110 005265 101 104 040 .ASCII /AD /
1111 005270 104 101 124 .ASCII /DAT/
1112 005273 101 072 040 .ASCII /A: /
1113 005276 045 117 066 .ASCII /X06/
1114 005301 045 116 045 .ASCII /XN%/
1115 005304 116 000 .ASCII /N/<00>
1116 005306 045 116 045 P.AAQ: .ASCII /XN%/
1117 005311 101 104 111 .ASCII /ADI/
1118 005314 101 107 116 .ASCII /AGN/
1119 005317 117 123 111 .ASCII /OSI/
1120 005322 116 107 040 .ASCII /NG /
1121 005325 125 116 111 .ASCII /UNI/
1122 005330 124 040 045 .ASCII /T %/
1123 005333 117 061 045 .ASCII /O1%/
1124 005336 116 045 116 .ASCII /N%N/
1125 005341 000 .ASCII <00>
1126 005342 045 101 124 P.AAR: .ASCII /XAT/
1127 005345 111 115 105 .ASCII /IME/
1128 005350 104 040 117 .ASCII /D O/
1129 005353 125 124 040 .ASCII /UT /
1130 005356 104 125 122 .ASCII /DUR/
1131 005361 111 116 107 .ASCII /ING/

```

```

1133      :ML4
1134      :
1135
1136 005364      040      115      102      .ASCII / MB/
1137 005367      125      123      040      .ASCII /US /
1138 005372      045      117      062      .ASCII /%02/
1139 005375      045      101      040      .ASCII /%A /
1140 005400      106      125      116      .ASCII /FUN/
1141 005403      103      045      116      .ASCII /C%N/
1142 005406      045      116      000      .ASCII /%N/<00>
1143 005411      000
1144 005412      045      101      103      P.AAS: .ASCII /%AC/
1145 005415      122      103      040      .ASCII /RC /
1146 005420      107      105      116      .ASCII /GEN/
1147 005423      040      075      040      .ASCII / = /
1148 005426      102      072      040      .ASCII /B: /
1149 005431      045      102      066      .ASCII /%B6/
1150 005434      045      101      040      .ASCII /%A /
1151 005437      101      072      040      .ASCII /A: /
1152 005442      045      102      066      .ASCII /%B6/
1153 005445      045      101      040      .ASCII /%A /
1154 005450      120      072      040      .ASCII /P: /
1155 005453      045      102      066      .ASCII /%B6/
1156 005456      045      116      000      .ASCII /%N/<00>
1157 005461      000
1158 005462      045      101      103      P.AAT: .ASCII /%AC/
1159 005465      122      103      040      .ASCII /RC /
1160 005470      103      101      114      .ASCII /CAL/
1161 005473      040      075      040      .ASCII / = /
1162 005476      102      072      040      .ASCII /B: /
1163 005501      045      112      066      .ASCII /%B6/
1164 005504      045      101      040      .ASCII /%A /
1165 005507      101      072      040      .ASCII /A: /
1166 005512      045      102      066      .ASCII /%B6/
1167 005515      045      101      040      .ASCII /%A /
1168 005520      120      072      040      .ASCII /P: /
1169 005523      045      102      066      .ASCII /%B6/
1170 005526      045      116      045      .ASCII /%N% /
1171 005531      116      000      000      .ASCII /N/<00><00>
1172 005534      045      101      106      P.AAU: .ASCII /%AF/
1173 005537      101      111      114      .ASCII /AIL/
1174 005542      105      104      040      .ASCII /ED /
1175 005545      101      124      040      .ASCII /AT /
1176 005550      120      114      117      .ASCII /PLO/
1177 005553      107      072      040      .ASCII /G: /
1178 005556      045      104      062      .ASCII /%D2/
1179 005561      045      101      040      .ASCII /%A /
1180 005564      103      110      101      .ASCII /CHA/
1181 005567      116      116      105      .ASCII /%NE/
1182 005572      114      072      040      .ASCII /L: /
1183 005575      045      104      062      .ASCII /%D2/
1184 005600      045      116      045      .ASCII /%N% /
1185 005603      116      000      000      .ASCII /N/<00><00>
1186 005606      045      101      106      P.AAV: .ASCII /%AF/
1187 005611      101      111      114      .ASCII /AIL/

```



```

1189          :ML4
1190          :
1191
1192 005614      105      104      040      .ASCII /ED /
1193 005617      101      124      040      .ASCII /AT /
1194 005622      127      122      104      .ASCII /WRD/
1195 005625      072      040      045      .ASCII /: %/
1196 005630      104      062      045      .ASCII /D2%/
1197 005633      101      040      102      .ASCII /A B/
1198 005636      111      124      072      .ASCII /IT:/
1199 005641      040      045      104      .ASCII / %D/
1200 005644      062      045      116      .ASCII /2%N/
1201 005647      045      116      000      .ASCII /%N/<00>
1202 005652      045      101      122      P.AAW: .ASCII /%AR/
1203 005655      105      107      111      .ASCII /EGI/
1204 005660      123      124      105      .ASCII /STE/
1205 005663      122      045      123      .ASCII /R%S/
1206 005666      063      045      101      .ASCII /3%A/
1207 005671      101      104      104      .ASCII /ADD/
1208 005674      122      105      123      .ASCII /RES/
1209 005677      123      045      123      .ASCII /S%S/
1210 005702      067      045      101      .ASCII /7%A/
1211 005705      103      117      116      .ASCII /CON/
1212 005710      124      105      116      .ASCII /TEN/
1213 005713      124      123      045      .ASCII /T%Z/
1214 005716      116      045      116      .ASCII /N%N/
1215 005721      000
1216 005722      045      123      045      P.AAX: .ASCII /%S%/
1217 005725      124      045      123      .ASCII /T%S/
1218 005730      065      045      117      .ASCII /5%O/
1219 005733      066      045      123      .ASCII /6%S/
1220 005736      064      045      117      .ASCII /4%O/
1221 005741      061      066      045      .ASCII /16%/
1222 005744      116      000
1223 005746      045      123      064      P.AAY: .ASCII /%S4/
1224 005751      045      124      045      .ASCII /%T%/
1225 005754      101      072      045      .ASCII /A: %/
1226 005757      123      045      117      .ASCII /S%O/
1227 005762      066      000
1228 005764      045      123      064      P.AAZ: .ASCII /%S4/
1229 005767      045      124      045      .ASCII /%T%/
1230 005772      101      072      045      .ASCII /A: %/
1231 005775      123      045      104      .ASCII /S%D/
1232 006000      061      045      104      .ASCII /1%D/
1233 006003      061      045      104      .ASCII /1%D/
1234 006006      061      045      104      .ASCII /1%D/
1235 006011      061      045      116      .ASCII /1%N/
1236 006014      000      000
1237 006016      045      124      045      P.ABA: .ASCII /%T%/
1238 006021      116      000      000      .ASCII /N/<00><00>
1239 006024      045      124      045      P.ABB: .ASCII /%T%/
1240 006027      124      045      116      .ASCII /T%N/
1241 006032      000      000
1242 006034      045      124      045      P.ABC: .ASCII /%T%/
1243 006037      124      045      124      .ASCII /T%T/

```

```

1245      ;ML4
1246      :
1247
1248 006042      045      116      000      .ASCII /%N/<00>
1249 006045      000      .ASCII <00>
1250 006046      045      124      045      P.ABD: .ASCII /%T%/
1251 006051      124      045      124      .ASCII /%T%/
1252 006054      045      124      045      .ASCII /%T%/
1253 006057      116      000      000      .ASCII /%N/<00><00>
1254 006062      045      124      045      P.ABE: .ASCII /%T%/
1255 006065      124      045      124      .ASCII /%T%/
1256 006070      045      124      045      .ASCII /%T%/
1257 006073      124      045      116      .ASCII /%N/
1258 006076      000      000      .ASCII <00><00>
1259 006100      045      124      045      P.ABF: .ASCII /%T%/
1260 006103      124      045      124      .ASCII /%T%/
1261 006106      045      124      045      .ASCII /%T%/
1262 006111      124      045      124      .ASCII /%T%/
1263 006114      045      116      000      .ASCII /%N/<00>
1264 006117      000      .ASCII <00>
1265 006120      045      124      045      P.ABG: .ASCII /%T%/
1266 006123      124      045      124      .ASCII /%T%/
1267 006126      045      124      045      .ASCII /%T%/
1268 006131      124      045      124      .ASCII /%T%/
1269 006134      045      124      045      .ASCII /%T%/
1270 006137      116      000      000      .ASCII /%N/<00><00>
1271 006142      045      124      045      P.ABH: .ASCII /%T%/
1272 006145      124      045      124      .ASCII /%T%/
1273 006150      045      124      045      .ASCII /%T%/
1274 006153      124      045      124      .ASCII /%T%/
1275 006156      045      124      045      .ASCII /%T%/
1276 006161      124      045      116      .ASCII /%N/
1277 006164      000      000      .ASCII <00><00>
1278 006166      045      124      045      P.ABI: .ASCII /%T%/
1279 006171      124      045      124      .ASCII /%T%/
1280 006174      045      124      045      .ASCII /%T%/
1281 006177      124      045      124      .ASCII /%T%/
1282 006202      045      124      045      .ASCII /%T%/
1283 006205      124      045      124      .ASCII /%T%/
1284 006210      045      116      000      .ASCII /%N/<00>
1285 006213      000      .ASCII <00>
1286 006214      045      124      045      P.ABJ: .ASCII /%T%/
1287 006217      124      045      124      .ASCII /%T%/
1288 006222      045      124      045      .ASCII /%T%/
1289 006225      124      045      124      .ASCII /%T%/
1290 006230      045      124      045      .ASCII /%T%/
1291 006233      124      045      124      .ASCII /%T%/
1292 006236      045      124      045      .ASCII /%T%/
1293 006241      116      000      000      .ASCII /%N/<00><00>
1294 006244      045      124      045      P.ABK: .ASCII /%T%/
1295 006247      124      045      124      .ASCII /%T%/
1296 006252      045      124      045      .ASCII /%T%/
1297 006255      124      045      124      .ASCII /%T%/
1298 006260      045      124      045      .ASCII /%T%/
1299 006263      124      045      124      .ASCII /%T%/

```

```
1301          ;ML4
1302          ;
1303
1304 006266    045    124    045          .ASCII /XTX/
1305 006271    124    045    116          .ASCII /TXN/
1306 006274    000    000          .ASCII <00><00>
1307 006276    040    107    117 P.ABL: .ASCII / GO/
1308 006301    000          .ASCII <00>
1309 006302    040    104    122 P.ABM: .ASCII / DR/
1310 006305    126    137    122          .ASCII /V R/
1311 006310    104    131    000          .ASCII /DY/<00>
1312 006313    000          .ASCII <00>
1313 006314    040    111    114 P.ABN: .ASCII / TL/
1314 006317    106    000    000          .ASCII /F/<00><00>
1315 006322    040    117    120 P.ABO: .ASCII / OP/
1316 006325    111    000    000          .ASCII /I/<00><00>
1317 006330    040    102    101 P.ABP: .ASCII / BA/
1318 006333    104    000    000          .ASCII /D/<00><00>
1319 006336    040    107    117 P.ABQ: .ASCII / GO/
1320 006341    117    104    000          .ASCII /OD/<00>
1321 006344    040    120    101 P.ABR: .ASCII / PA/
1322 006347    122    111    124          .ASCII /RIT/
1323 006352    131    040    116          .ASCII /Y N/
1324 006355    117    124    000          .ASCII /OT/<00>
1325 006360    040    107    105 P.ABS: .ASCII / GE/
1326 006363    116    105    122          .ASCII /NER/
1327 006366    101    124    105          .ASCII /ATE/
1328 006371    104    000    000          .ASCII /D/<00><00>
1329 006374    040    104    105 P.ABT: .ASCII / DE/
1330 006377    124    105    103          .ASCII /TEC/
1331 006402    124    105    104          .ASCII /TED/
1332 006405    000          .ASCII <00>
1333 006406    040    105    122 P.ABU: .ASCII / ER/
1334 006411    122    117    122          .ASCII /ROR/
1335 006414    000    000          .ASCII <00><00>
1336 006416    040    101    106 P.ABV: .ASCII / AF/
1337 006421    124    105    122          .ASCII /TER/
1338 006424    000    000          .ASCII <00><00>
1339 006426    040    104    125 P.ABW: .ASCII / DU/
1340 006431    122    111    116          .ASCII /RIN/
1341 006434    107    000          .ASCII /G/<00>
1342 006436    040    101    124 P.ABX: .ASCII / AT/
1343 006441    000          .ASCII <00>
1344 006442    040    106    101 P.ABY: .ASCII / FA/
1345 006445    111    114    125          .ASCII /ILU/
1346 006450    122    105    000          .ASCII /RE/<00>
1347 006453    000          .ASCII <00>
1348 006454    040    101    124 P.ABZ: .ASCII / AT/
1349 006457    101    000    000          .ASCII /A/<00><00>
1350 006462    040    101    124 P.ACA: .ASCII / AT/
1351 006465    124    116    000          .ASCII /TN/<00>
1352 006470    040    127    122 P.ACB: .ASCII / WR/
1353 006473    111    124    111          .ASCII /ITI/
1354 006476    116    107    000          .ASCII /NG/<00>
1355 006501    000          .ASCII <00>
```

```

1357      :ML4
1358      :
1359
1360 006502 040 126 126 P.ACC: .ASCII /VV/
1361 006505 000      .ASCII <00>
1362 006506 040 106 125 P.ACD: .ASCII /FU/
1363 006511 116 103 000      .ASCII /NC/<00>
1364 006514 040 124 122 P.ACE: .ASCII /TR/
1365 006517 105 000 000      .ASCII /E/<00><00>
1366 006522 040 122 115 P.ACF: .ASCII /RM/
1367 006525 122 000 000      .ASCII /R/<00><00>
1368 006530 040 105 130 P.ACG: .ASCII /EX/
1369 006533 103 105 123      .ASCII /CES/
1370 006536 123 111 126      .ASCII /SIV/
1371 006541 105 000 000      .ASCII /E/<00><00>
1372 006544 040 115 102 P.ACH: .ASCII /MB/
1373 006547 125 123 000      .ASCII /US/<00>
1374 006552 040 104 101 P.ACI: .ASCII /DA/
1375 006555 124 101 000      .ASCII /TA/<00>
1376 006560 040 103 117 P.ACJ: .ASCII /CO/
1377 006563 116 124 111      .ASCII /NTI/
1378 006566 116 125 111      .ASCII /NUI/
1379 006571 124 131 000      .ASCII /TY/<00>
1380 006574 040 101 117 P.ACK: .ASCII /AO/
1381 006577 105 000 000      .ASCII /E/<00><00>
1382 006602 040 114 102 P.ACL: .ASCII /LB/
1383 006605 124 000 000      .ASCII /T/<00><00>
1384 006610 040 120 122 P.ACM: .ASCII /PR/
1385 006613 105 115 101      .ASCII /EMA/
1386 006616 124 125 122      .ASCII /TUR/
1387 006621 114 131 000      .ASCII /LY/<00>
1388 006624 040 111 101 P.ACN: .ASCII /IA/
1389 006627 105 000 000      .ASCII /E/<00><00>
1390 006632 040 111 116 P.ACO: .ASCII /IN/
1391 006635 103 122 105      .ASCII /CRE/
1392 006640 115 105 116      .ASCII /MEN/
1393 006643 124 000 000      .ASCII /T/<00><00>
1394 006646 040 127 111 P.ACP: .ASCII /WI/
1395 006651 124 110 000      .ASCII /TH/<00>
1396 006654 040 125 126 P.ACQ: .ASCII /UV/
1397 006657 000      .ASCII <00>
1398 006660 040 125 116 P.ACR: .ASCII /UN/
1399 006663 123 000 000      .ASCII /S/<00><00>
1400 006666 040 120 122 P.ACS: .ASCII /PR/
1401 006671 117 115 000      .ASCII /OM/<00>
1402 006674 040 117 122 P.ACT: .ASCII /OR/
1403 006677 000      .ASCII <00>
1404 006700 040 123 105 P.ACU: .ASCII /SE/
1405 006703 114 105 103      .ASCII /LEC/
1406 006706 124 000      .ASCII /T/<00>
1407 006710 040 122 105 P.ACV: .ASCII /RE/
1408 006713 107 000 000      .ASCII /G/<00><00>
1409 006716 040 125 116 P.ACW: .ASCII /UN/
1410 006721 111 121 125      .ASCII /IQU/
1411 006724 105 000      .ASCII /E/<00>

```

```

1413      :ML4
1414      :
1415
1416 006726 040 061 064 P.ACX: .ASCII / 14/
1417 006731 000      .ASCII <00>
1418 006732 040 116 111 P.ACY: .ASCII / NI/
1419 006735 102 102 114 .ASCII /BBL/
1420 006740 105 040 103 .ASCII /E C/ - - -
1421 006743 116 124 000 .ASCII /N?/<00>
1422 006746 040 107 124 P.ACZ: .ASCII / GT/
1423 006751 122 000 000 .ASCII /R/<00><00>
1424 006754 040 127 110 P.ADA: .ASCII / WH/
1425 006757 111 114 105 .ASCII /I'E/
1426 006762 000 000 .ASCII <00><00>
1427 006764 040 124 122 P.ADB: .ASCII / TR/
1428 006767 105 000 000 .ASCII /E/<00><00>
1429 006772 040 111 116 P.ADC: .ASCII / IN/
1430 006775 111 124 111 .ASCII /ITI/
1431 007000 101 114 000 .ASCII /AL/<00>
1432 007003 000      .ASCII <00>
1433 007004 040 117 106 P.ADD: .ASCII / OF/
1434 007007 106 137 123 .ASCII /F S/
1435 007012 105 124 000 .ASCII /ET/<00>
1436 007015 000      .ASCII <00>
1437 007016 040 103 117 P.ADE: .ASCII / CO/
1438 007021 125 116 124 .ASCII /UNT/
1439 007024 000 000 .ASCII <00><00>
1440 007026 040 104 105 P.ADF: .ASCII / DE/
1441 007031 114 101 131 .ASCII /LAY/
1442 007034 000 000 .ASCII <00><00>
1443 007036 040 124 105 P.ADG: .ASCII / TE/
1444 007041 123 124 123 .ASCII /STS/
1445 007044 000 000 .ASCII <00><00>
1446 007046 040 101 104 P.ADH: .ASCII / AD/
1447 007051 122 123 000 .ASCII /RS/<00>
1448 007054 040 103 117 P.ADI: .ASCII / CO/
1449 007057 125 116 124 .ASCII /UNT/
1450 007062 105 122 000 .ASCII /ER/<00>
1451 007065 000      .ASCII <00>
1452 007066 040 122 105 P.ADJ: .ASCII / RE/
1453 007071 107 000 000 .ASCII /G/<00><00>
1454 007074 040 124 105 P.ADK: .ASCII / TE/
1455 007077 123 124 105 .ASCII /STE/
1456 007102 104 060 .ASCII /D/<00>
1457 007104 040 116 111 P.ADL: .ASCII / NI/
1458 007107 102 102 114 .ASCII /BBL/
1459 007112 105 000 .ASCII /E/<00>
1460 007114 040 101 114 P.ADM: .ASCII / AL/
1461 007117 114 000 000 .ASCII /L/<00><00>
1462 007122 040 124 105 P.ADN: .ASCII / TE/
1463 007125 123 124 000 .ASCII /ST/<00>
1464 007130 040 130 106 P.ADO: .ASCII / XF/
1465 007133 105 122 105 .ASCII /ERE/
1466 007136 104 000 .ASCII /D/<00>
1467 007140 040 116 111 P.ADP: .ASCII / NI/

```

```

1469          :ML4
1470          :
1471
1472 007143    102    102    114          .ASCII /BBL/
1473 007146    105    123    000          .ASCII /ES/<00>
1474 007151    000          .ASCII <00>
1475 007152    040    123    103 P.ADQ: .ASCII / SC/
1476 007155    000          .ASCII <00>
1477 007156    040    115    125 P.ADR: .ASCII / MU/
1478 007161    114    124    111          .ASCII /LTI/
1479 007164    120    114    105          .ASCII /PLE/
1480 007167    130    105    122          .ASCII /XER/
1481 007172    000    000          .ASCII <00><00>
1482 007174    040    125    116 P.ADS: .ASCII / UN/
1483 007177    105    130    120          .ASCII /EXP/
1484 007202    105    103    124          .ASCII /ECT/
1485 007205    105    104    000          .ASCII /ED/<00>
1486 007210    040    116    105 P.ADT: .ASCII / NE/
1487 007213    104    000    000          .ASCII /D/<00><00>
1488 007216    040    111    114 P.ADU: .ASCII / IL/
1489 007221    122    000    000          .ASCII /R/<00><00>
1490 007224    040    103    122 P.ADV: .ASCII / CR/
1491 007227    103    000    000          .ASCII /C/<00><00>
1492 007232    040    123    107 P.ADW: .ASCII / SG/
1493 007235    114    000    000          .ASCII /L/<00><00>
1494 007240    040    105    103 P.ADX: .ASCII / EC/
1495 007243    110    000    000          .ASCII /H/<00><00>
1496 007246    125    116    103 P.ADY: .ASCII /UNC/
1497 007251    000          .ASCII <00>
1498 007252    040    102    111 P.ADZ: .ASCII / BI/
1499 007255    124    000    000          .ASCII /T/<00><00>
1500 007260    040    103    110 P.AEA: .ASCII / CH/
1501 007263    101    116    116          .ASCII /ANV/
1502 007266    105    114    000          .ASCII /EL/<00>
1503 007271    000          .ASCII <00>
1504 007272    040    114    101 P.AEB: .ASCII / LA/
1505 007275    124    103    110          .ASCII /TCH/
1506 007300    000    000          .ASCII <00><00>
1507 007302    040    104    103 P.AEC: .ASCII / DC/
1508 007305    113    000    000          .ASCII /K/<00><00>
1509 007310    040    102    125 P.AED: .ASCII / BU/
1510 007313    123    000    000          .ASCII /S/<00><00>
1511 007316    040    123    131 P.AEE: .ASCII / SY/
1512 007321    116    104    122          .ASCII /NDR/
1513 007324    117    116    105          .ASCII /ONE/
1514 007327    000          .ASCII <00>
1515 007330    040    104    105 P.AEF: .ASCII / DE/
1516 007333    103    117    104          .ASCII /COD/
1517 007336    105    000          .ASCII /E/<00>
1518 007340    040    116    117 P.AEG: .ASCII / NO/
1519 007343    124    000    000          .ASCII /T/<00><00>
1520 007346    040    125    116 P.AEH: .ASCII / UN/
1521 007351    103    117    122          .ASCII /COR/
1522 007354    122    105    103          .ASCII /REC/
1523 007357    124    101    102          .ASCII /TAB/

```

```

1525      :ML4
1526      :
1527
1528 007362 114 105 000      .ASCII /LE/<00>
1529 007365 000          .ASCII <00>
1530 007366 040 102 111 P.AEI: .ASCII / BI/
1531 007371 124 040 116      .ASCII /T N/
1532 007374 117 124 040      .ASCII /OT /
1533 007377 123 105 124      .ASCII /SET/
1534 007402 000 000          .ASCII <00><00>
1535 007404 040 102 111 P.AEJ: .ASCII / BI/
1536 007407 124 040 116      .ASCII /T N/
1537 007412 117 124 040      .ASCII /OT /
1538 007415 103 114 122      .ASCII /CLR/
1539 007420 000 000          .ASCII <00><00>
1540 007422 040 116 117 P.AEK: .ASCII / NO/
1541 007425 040 122 105      .ASCII / RE/
1542 007430 123 120 117      .ASCII /SPO/
1543 007433 116 103 105      .ASCII /NCE/
1544 007436 040 101 106      .ASCII / AF/
1545 007441 124 105 122      .ASCII /TER/
1546 007444 040 061 056      .ASCII / 1./
1547 007447 065 040 125      .ASCII /5 U/
1548 007452 123 000          .ASCII /S/<00>
1549 007454 040 104 101 P.AEL: .ASCII / DA/
1550 007457 124 101 040      .ASCII /TA /
1551 007462 105 122 122      .ASCII /ERR/
1552 007465 117 122 123      .ASCII /ORS/
1553 007470 000 000          .ASCII <00><00>
1554 007472 040 102 111 P.AEM: .ASCII / BI/
1555 007475 124 040 123      .ASCII /T S/
1556 007500 105 124 000      .ASCII /ET/<00>
1557 007503 000          .ASCII <00>
1558 007504 040 102 111 P.AEN: .ASCII / BI/
1559 007507 124 040 103      .ASCII /T C/
1560 007512 114 122 000      .ASCII /LR/<00>
1561 007515 000          .ASCII <00>
1562 007516 040 117 106 P.AEO: .ASCII / OF/
1563 007521 040 117 124      .ASCII / OT/
1564 007524 110 105 122      .ASCII /HER/
1565 007527 040 104 122      .ASCII / DR/
1566 007532 111 126 105      .ASCII /IVE/
1567 007535 123 000 000      .ASCII /S/<00><00>
1568 007540 040 103 114 P.AEP: .ASCII / CL/
1569 007543 101 123 123      .ASCII /ASS/
1570 007546 040 101 000      .ASCII / A/<00>
1571 007551 000          .ASCII <00>
1572 007552 040 103 114 P.AEQ: .ASCII / CL/
1573 007555 101 123 123      .ASCII /ASS/
1574 007560 040 102 000      .ASCII / B/<00>
1575 007563 000          .ASCII <00>
1576 007564 040 124 117 P.AER: .ASCII / TO/
1577 007567 040 106 111      .ASCII / FI/
1578 007572 116 104 000      .ASCII /ND/<00>
1579 007575 000          .ASCII <00>

```

```

1581      ;ML4
1582      ;
1583
1584 007576 040 116 117 P.AES: .ASCII / NO/
1585 007601 124 040 114      .ASCII /T L/
1586 007604 101 124 103      .ASCII /ATC/
1587 007607 110 105 104      .ASCII /HED/
1588 007612 000 000      .ASCII <00><00>
1589 007614 040 123 111 P.AET: .ASCII / SI/
1590 007617 116 107 114      .ASCII /NGL/
1591 007622 105 040 102      .ASCII /E B/
1592 007625 111 124 040      .ASCII /IT /
1593 007630 105 122 122      .ASCII /ERR/
1594 007633 117 122 000      .ASCII /OR/<00>
1595 007636 040 115 125 P.AEU: .ASCII / MU/
1596 007641 114 124 111      .ASCII /LTI/
1597 007644 120 114 105      .ASCII /PLE/
1598 007647 040 102 111      .ASCII / BI/
1599 007652 124 040 105      .ASCII /T E/
1600 007655 122 122 117      .ASCII /RRO/
1601 007660 122 000      .ASCII /R/<00>
1602 007662 040 040 040 P.AEV: .ASCII / /
1603 007665 040 040 040      .ASCII / /
1604 007670 040 040 040      .ASCII / /
1605 007673 122 105 107      .ASCII /REG/
1606 007676 111 123 124      .ASCII /IST/
1607 007701 105 122 040      .ASCII /ER /
1608 007704 040 040 104      .ASCII / D/
1609 007707 125 115 120      .ASCII /UMP/
1610 007712 000 000      .ASCII <00><00>
1611 007714 123 105 122 P.AEW: .ASCII /SER/
1612 007717 111 101 114      .ASCII /IAL/
1613 007722 040 043 000      .ASCII / #/<00>
1614 007725 000      .ASCII <00>
1615 007726 040 115 105 P.AEX: .ASCII / ME/
1616 007731 115 040 123      .ASCII /M S/
1617 007734 111 132 111      .ASCII /IZI/
1618 007737 116 107 000      .ASCII /NG/<00>
1619 007742 040 116 117 P.AEY: .ASCII / NO/
1620 007745 117 120 000      .ASCII /OP/<00>
1621 007750 040 104 122 P.AEZ: .ASCII / DR/
1622 007753 126 000 000      .ASCII /V/<00><00>
1623 007756 040 127 122 P.AFA: .ASCII / WR/
1624 007761 111 124 105      .ASCII /ITE/
1625 007764 040 103 110      .ASCII / CH/
1626 007767 105 103 113      .ASCII /ECK/
1627 007772 000 000      .ASCII <00><00>
1628 007774 040 127 122 P.AFB: .ASCII / WR/
1629 007777 111 124 105      .ASCII /ITE/
1630 010002 000 000      .ASCII <00><00>
1631 010004 040 122 105 P.AFC: .ASCII / RE/
1632 010007 101 104 000      .ASCII /AD/<00>
1633 010012 040 103 114 P.AFD: .ASCII / CL/
1634 010015 105 101 122      .ASCII /EAR/
1635 010020 000 000      .ASCII <00><00>

```



```

1637      ;ML4
1638      ;
1639
1640 010022 040 103 117 P.AFE: .ASCII / CO/
1641 010025 115 120 040      .ASCII /MP /
1642 010030 105 122 122      .ASCII /ERR/
1643 010033 117 122 000      .ASCII /OR/<00>
1644 010036 040 123 131 P.AFF: .ASCII / SY/
1645 010041 123 040 103      .ASCII /S C/
1646 010044 114 122 000      .ASCII /LR/<00>
1647 010047 090      .ASCII <00>
1648 010050 040 123 105 P.AFG: .ASCII / SE/
1649 010053 101 122 103      .ASCII /ARC/
1650 010056 110 000      .ASCII /H/<00>
1651 010060 040 122 105 P.AFH: .ASCII / RE/
1652 010063 101 104 055      .ASCII /AD-/
1653 010066 111 116 055      .ASCII /IN-/
1654 010071 120 122 105      .ASCII /PRE/
1655 010074 123 105 124      .ASCII /SET/
1656 010077 000      .ASCII <00>
1657 010100 040 111 114 P.AFI: .ASCII / IL/
1658 010103 114 105 107      .ASCII /LEG/
1659 010106 101 114 000      .ASCII /AL/<00>
1660 010111 000      .ASCII <00>
1661 010112 040 101 102 P.AFJ: .ASCII / AB/
1662 010115 117 122 124      .ASCII /ORT/
1663 010120 000 000      .ASCII <00><00>
1664 010122 040 101 122 P.AFK: .ASCII / AR/
1665 010125 122 040 122      .ASCII /R R/
1666 010130 104 137 127      .ASCII /D W/
1667 010133 122 124 000      .ASCII /RT/<00>
1668 010136 040 107 117 P.AFL: .ASCII / GO/
1669 010141 117 104 040      .ASCII /OD /
1670 010144 102 114 113      .ASCII /BLK/
1671 010147 000      .ASCII <00>
1672 010150 040 122 105 P.AFM: .ASCII / RE/
1673 010153 106 122 105      .ASCII /FRE/
1674 010156 123 110 000      .ASCII /SH/<00>
1675 010161 000      .ASCII <00>
1676 010162 040 101 122 P.AFN: .ASCII / AR/
1677 010165 122 101 131      .ASCII /RAY/
1678 010170 000 000      .ASCII <00><00>
1679 010172 040 122 101 P.AFO: .ASCII / RA/
1680 010175 115 055 102      .ASCII /M-B/
1681 010200 125 123 000      .ASCII /US/<00>
1682 010203 000      .ASCII <00>
1683 010204 040 117 126 P.AFP: .ASCII / OV/
1684 010207 105 122 106      .ASCII /ERF/
1685 010212 114 117 127      .ASCII /LOW/
1686 010215 000      .ASCII <00>
1687 010216 040 103 110 P.AFQ: .ASCII / CH/
1688 010221 113 137 123      .ASCII /K S/
1689 010224 125 115 000      .ASCII /UM/<00>
1690 010227 000      .ASCII <00>
1691 010230 040 114 101 P.AFR: .ASCII / LA/

```

```

1693          :ML4
1694          :
1695
1696 010233    123    124    040          .ASCII /ST /
1697 010236    102    114    113          .ASCII /BLK/
1698 010241    000          .ASCII <00>
1699 010242    040    111    116 P.AFS: .ASCII / IN/
1700 010245    111    124    111          .ASCII /ITI/
1701 010250    101    114    111          .ASCII /ALI/
1702 010253    132    105    000          .ASCII /ZE/<00>
1703 010256    040    115    114 P.AFT: .ASCII / ML/
1704 010261    103    123    061          .ASCII /CS1/
1705 010264    000    000          .ASCII <00><00>
1706 010266    040    115    114 P.AFU: .ASCII / ML/
1707 010271    104    123    000          .ASCII /DS/<00>
1708 010274    040    115    114 P.AFV: .ASCII / ML/
1709 010277    105    122    000          .ASCII /ER/<00>
1710 010302    040    115    114 P.AFW: .ASCII / ML/
1711 010305    115    122    000          .ASCII /MR/<00>
1712 010310    040    115    114 P.AFX: .ASCII / ML/
1713 010313    101    123    000          .ASCII /AS/<00>
1714 010316    040    115    114 P.AFY: .ASCII / ML/
1715 010321    104    101    000          .ASCII /DA/<00>
1716 010324    040    115    114 P.AFZ: .ASCII / ML/
1717 010327    104    124    000          .ASCII /DT/<00>
1718 010332    040    115    114 P.AGA: .ASCII / ML/
1719 010335    120    101    000          .ASCII /PA/<00>
1720 010340    040    115    114 P.AGB: .ASCII / ML/
1721 010343    123    116    000          .ASCII /SN/<00>
1722 010346    040    115    114 P.AGC: .ASCII / ML/
1723 010351    105    061    000          .ASCII /E1/<00>
1724 010354    040    115    114 P.AGD: .ASCII / ML/
1725 010357    105    062    000          .ASCII /E2/<00>
1726 010362    040    115    114 P.AGE: .ASCII / ML/
1727 010365    104    061    000          .ASCII /D1/<00>
1728 010370    040    115    114 P.AGF: .ASCII / ML/
1729 010373    104    062    000          .ASCII /D2/<00>
1730 010376    040    115    114 P.AGG: .ASCII / ML/
1731 010401    105    105    000          .ASCII /EE/<00>
1732 010404    040    115    114 P.AGH: .ASCII / ML/
1733 010407    105    114    000          .ASCII /EL/<00>
1734 010412    040    115    114 P.AGI: .ASCII / ML/
1735 010415    120    104    000          .ASCII /PD/<00>
1736 010420    040    115    114 P.AGJ: .ASCII / ML/
1737 010423    103    123    062          .ASCII /CS2/
1738 010426    000    000          .ASCII <00><00>
1739 010430    040    115    114 P.AGK: .ASCII / ML/
1740 010433    127    103    000          .ASCII /WC/<00>
1741 010436    040    115    114 P.AGL: .ASCII / ML/
1742 010441    102    101    000          .ASCII /BA/<00>
1743 010444    040    115    114 P.AGM: .ASCII / ML/
1744 010447    102    101    111          .ASCII /BAI/
1745 010452    000    000          .ASCII <00><00>
1746 010454    040    115    114 P.AGN: .ASCII / ML/
1747 010457    103    123    063          .ASCII /CS3/

```

```

1749          :ML4
1750          :
1751
1752 010462      000      000
1753 010464      101      123      131 P.AGO: .ASCII <00><00>
1754 010467      116      103      110      .ASCII /ASY/
1755 010472      122      117      116      .ASCII /NCH/
1756 010475      117      125      123      .ASCII /RON/
1757 010500      040      106      101      .ASCII /OUS/
1758 010503      111      114      125      .ASCII / FA/
1759 010506      122      105      040      .ASCII /ILU/
1760 010511      115      117      104      .ASCII /RE /
1761 010514      125      114      105      .ASCII /MOD/
1762 010517      040      067      063      .ASCII /ULE/
1763 010522      066      061      000      .ASCII / 73/
1764 010525      000
1765 010526      123      131      116 P.AGP: .ASCII /61/<00>
1766 010531      103      110      122      .ASCII <00>
1767 010534      117      116      117      .ASCII /SYN/
1768 010537      125      123      040      .ASCII /CHR/
1769 010542      106      101      111      .ASCII /ONO/
1770 010545      114      125      122      .ASCII /US /
1771 010550      105      040      115      .ASCII /FAI/
1772 010553      117      104      125      .ASCII /LUR/
1773 010556      114      105      040      .ASCII /E M/
1774 010561      067      063      066      .ASCII /ODU/
1775 010564      062      040      000      .ASCII /LE /
1776 010567      000
1777 010570      101      122      122 P.AGQ: .ASCII /736/
1778 010573      101      131      040      .ASCII /2 /<00>
1779 010576      104      101      124      .ASCII <00>
1780 010601      101      040      106      .ASCII /ARR/
1781 010604      101      111      114      .ASCII /AY /
1782 010607      125      122      105      .ASCII /DAT/
1783 010612      040      115      117      .ASCII /A F/
1784 010615      104      125      114      .ASCII /AIL/
1785 010620      105      040      067      .ASCII /URE/
1786 010623      063      066      063      .ASCII / MO/
1787 010626      040      000
1788 010630      115      105      115 P.AGR: .ASCII /DUL/
1789 010633      117      122      131      .ASCII /E 7/
1790 010636      040      101      122      .ASCII /363/
1791 010641      122      101      131      .ASCII / /<00>
1792 010644      040      106      101      .ASCII /MEM/
1793 010647      111      114      125      .ASCII /ORY/
1794 010652      122      105      040      .ASCII / AR/
1795 010655      115      117      104      .ASCII /RAY/
1796 010660      125      114      105      .ASCII / FA/
1797 010663      040      067      063      .ASCII /ILU/
1798 010666      065      067      000      .ASCII /RE /
1799 010671      000
1800 010672      111      116      124 P.AGS: .ASCII /MOD/
1801 010675      105      122      115      .ASCII /ULE/
1802 010700      105      104      111      .ASCII / 73/
1803 010703      101      124      105      .ASCII /57/<00>

```

```

1805      :ML4
1806      :
1807      :
1808 010706      040      104      111      .ASCII / DI/
1809 010711      101      107      116      .ASCII /AGN/
1810 010714      117      123      124      .ASCII /OST/
1811 010717      111      103      040      .ASCII /IC /
1812 010722      115      105      123      .ASCII /MES/
1813 010725      123      101      107      .ASCII /SAG/
1814 010730      105      000      .ASCII /E/<00>
1815 010732      104      101      124 P.AGT: .ASCII /DAT/
1816 010735      101      040      114      .ASCII /A L/
1817 010740      101      124      105      .ASCII /ATE/
1818 010743      040      105      122      .ASCII / ER/
1819 010746      122      117      122      .ASCII /ROR/
1820 010751      040      104      125      .ASCII / DU/
1821 010754      122      111      116      .ASCII /RIN/
1822 010757      107      040      124      .ASCII /G T/
1823 010762      122      101      116      .ASCII /RAW/
1824 010765      123      106      105      .ASCII /SFE/
1825 010770      122      000      .ASCII /R/<00>
1826 010772      123      103      040 P.AGU: .ASCII /SC /
1827 010775      102      111      124      .ASCII /BIT/
1828 011000      040      123      105      .ASCII / SE/
1829 011003      124      040      104      .ASCII /T D/
1830 011006      125      122      111      .ASCII /URI/
1831 011011      116      107      040      .ASCII /NG /
1832 011014      124      122      101      .ASCII /TRA/
1833 011017      116      123      106      .ASCII /NSF/
1834 011022      105      122      000      .ASCII /ER/<00>
1835 011025      000      .ASCII <00>
1836 011026      124      122      117 P.AGV: .ASCII /TRO/
1837 011031      125      102      114      .ASCII /UBL/
1838 011034      105      040      123      .ASCII /E S/
1839 011037      110      117      117      .ASCII /HOO/
1840 011042      124      040      114      .ASCII /T L/
1841 011045      117      117      120      .ASCII /OOP/
1842 011050      040      105      122      .ASCII / ER/
1843 011053      122      117      122      .ASCII /ROR/
1844 011056      123      000      .ASCII /S/<00>
1845 011060      122      110      040 P.AGW: .ASCII /RH /
1846 011063      103      117      116      .ASCII /CON/
1847 011066      124      122      117      .ASCII /TRO/
1848 011071      114      114      105      .ASCII /LLE/
1849 011074      122      040      105      .ASCII /R E/
1850 011077      122      122      117      .ASCII /RRO/
1851 011102      122      123      000      .ASCII /RS/<00>
1852 011105      000      .ASCII <00>
1853 011106      104      122      111 P.AGX: .ASCII /DRI/
1854 011111      126      105      040      .ASCII /VE /
1855 011114      110      125      116      .ASCII /HUN/
1856 011117      107      040      101      .ASCII /G A/
1857 011122      106      124      105      .ASCII /FTE/
1858 011125      122      040      115      .ASCII /R MV/
1859 011130      101      123      123      .ASCII /ASS/

```

```

1861          :ML4
1862          :
1863
1864 011133    040    102    125    .ASCII /BU/
1865 011136    123    040    124    .ASCII /S T/
1866 011141    122    101    116    .ASCII /RAN/
1867 011144    123    106    105    .ASCII /SFE/
1868 011147    122    000    000    .ASCII /R/<00><00>
1869
1870
1871
1872 011152          NIB.SAVE:
1873 011152          .BLKW 3
1874 011160          HW.OR.TBL:
1875 011160          .BLKW 177
1876 011556          PTBL.PTR:
1877 011556          .BLKW 1
1878 011560          OP.NUM.ARR:
1879 011560          .BLKW 1
1880 011562          ARR.INC:.BLKW 1
1881 011564          GOOD.BLK:
1882 011564          .BLKW 1
1883 011566          PAR.DIS:.BLKW 1
1884 011570          CHIP.SIZ:
1885 011570          .BLKW 1
1886 011572          LST.BLK:.BLKW 1
1887 011574          ARR.16: .BLKW 1
1888 011576          LST.ARR:.BLKW 1
1889 011600          IO.BUF: .BLKW 400
1890 012600          STK.OFF:.BLKW 5
1891 012612          STACK: .BLKW 143
1892 013120          PD.TEMP:.BLKW 1
1893 013122          W.C.SIZE:
1894 013122          .BLKW 1
1895 013124          RAS.INC:.BLKW 1
1896 013126          WT.DATA:.BLKW 1
1897 013130          RD.DATA:.BLKW 1
1898 013132          DRIVE.TYPE:
1899 013132          .BLKW 1
1900 013134          LST.DUT.REG:
1901 013134          .BLKW 1
1902 013136          REG.INIT.FLG:
1903 013136          .WORD 0
1904 013140          A.CAL: .BLKW 1
1905 013142          B.CAL: .BLKW 1
1906 013144          P.CAL: .BLKW 1
1907 013146          A.GEN: .BLKW 1
1908 013150          B.GEN: .BLKW 1
1909 013152          P.GEN: .BLKW 1
1910 013154          ML.REG: .BYTE 0
1911 013155          .BYTE 0
1912 013156          .WORD 4000
1913 013160          .WORD -4077
1914 013162          .WORD -17600

```

000000

000

000

004000

173701

160200

```
1916  
1917  
1918  
1919 013164 000 .BYTE 0  
1920 013165 000 .BYTE 0  
1921 013166 000 .BYTE 0  
1922 013167 000 .BYTE 0  
1923 013170 000 .BYTE 0  
1924 013171 000 .BYTE 0  
1925 013172 000 .BYTE 0  
1926 013173 000 .BYTE 0  
1927 013174 000 .BYTE 0  
1928 013175 000 .BYTE 0  
1929 013176 000 .BYTE 0  
1930 013177 000 .BYTE 0  
1931 013200 000 .BYTE 0  
1932 013201 000 .BYTE 0  
1933 013202 000 .BYTE 0  
1934 013203 000 .BYTE 0  
1935 013204 000 .BYTE 0  
1936 013205 000 .BYTE 0  
1937 013206 000 .BYTE 0  
1938 013207 000 .BYTE 0  
1939 013210 000 .BYTE 0  
1940 013211 000 .BYTE 0  
1941 013212 000 .BYTE 0  
1942 013213 000 .BYTE 0  
1943 013214 000 .BYTE 0  
1944 013215 000 .BYTE 0  
1945 013216 000 .BYTE 0  
1946 013217 000 .BYTE 0  
1947 013220 000 .BYTE 0  
1948 013221 000 .BYTE 0  
1949 013222 000 .BYTE 0  
1950 013223 000 .BYTE 0  
1951 013224 000 .BYTE 0  
1952 013225 000 .BYTE 0  
1953 013226 010600 .WORD 10600  
1954 013230 025077 .WORD 25077  
1955 013232 000100 .WORD 100  
1956 013234 000 .BYTE 0  
1957 013235 000 .BYTE 0  
1958 013236 000 .BYTE 0  
1959 013237 000 .BYTE 0  
1960 013240 014620 .WORD 14620  
1961 013242 000 .BYTE 0  
1962 013243 000 .BYTE 0  
1963 013244 000 .BYTE 0  
1964 013245 000 .BYTE 0  
1965 013246 000 .BYTE 0  
1966 013247 000 .BYTE 0  
1967 013250 000 .BYTE 0  
1968 013251 000 .BYTE 0  
1969 013252 177400 .WORD -400  
1970 013254 000 .BYTE 0
```

:ML4

```
1972  
1973  
1974  
1975 013255 000 .BYTE 0  
1976 013256 000 .BYTE 0  
1977 013257 000 .BYTE 0  
1978 013260 100000 .WORD -100000  
1979 013262 000 .BYTE 0  
1980 013263 000 .BYTE 0  
1981 013264 000 .BYTE 0  
1982 013265 000 .BYTE 0  
1983 013266 000 .BYTE 0  
1984 013267 000 .BYTE 0  
1985 013270 000 .BYTE 0  
1986 013271 000 .BYTE 0  
1987 013272 000 .BYTE 0  
1988 013273 000 .BYTE 0  
1989 013274 000 .BYTE 0  
199 013275 000 .BYTE 0  
1991 013276 000 .BYTE 0  
1992 013277 000 .BYTE 0  
1993 013300 00020 .WORD 20  
1994 013302 177400 .WORD -400  
1995 013304 000 .BYTE 0  
1996 013305 000 .BYTE 0  
1997 013306 000110 .WORD 110  
1998 013310 177666 .WORD -112  
1999 013312 000001 .WORD 1  
2000 013314 000 .BYTE 0  
2001 013315 000 .BYTE 0  
2002 013316 000 .BYTE 0  
2003 013317 000 .BYTE 0  
2004 013320 000 .BYTE 0  
2005 013321 000 .BYTE 0  
2006 013322 000 .BYTE 0  
2007 013323 000 .BYTE 0  
2008 013324 000 .BYTE 0  
2009 013325 000 .BYTE 0  
2010 013326 000 .BYTE 0  
2011 013327 000 .BYTE 0  
2012 013330 140300 .WORD -37500  
2013 013332 000 .BYTE 0  
2014 013333 000 .BYTE 0  
2015 013334 000 .BYTE 0  
2016 013335 000 .BYTE 0  
2017 013336 000 .BYTE 0  
2018 013337 000 .BYTE 0  
2019 013340 100300 .WORD -77500  
2020 013342 000 .BYTE 0  
2021 013343 000 .BYTE 0  
2022 013344 000 .BYTE 0  
2023 013345 000 .BYTE 0  
2024 013346 000 .BYTE 0  
2025 013347 000 .BYTE 0  
2026 013350 000 .BYTE 0
```

:ML4
:

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

```
2028      :ML4
2029      :
2030
2031 013351      000      .BYTE      0
2032 013352      000      .BYTE      0
2033 013353      000      .BYTE      0
2034 013354      000      .BYTE      0
2035 013355      000      .BYTE      0
2036 013356      000      .BYTE      0
2037 013357      000      .BYTE      0
2038 013360      000      .BYTE      0
2039 013361      000      .BYTE      0
2040 013362      000      .BYTE      0
2041 013363      000      .BYTE      0
2042 013364      000      .BYTE      0
2043 013365      000      .BYTE      0
2044 013366      000      .BYTE      0
2045 013367      000      .BYTE      0
2046 013370      010000    .WORD     10000
2047 013372      000      .BYTE      0
2048 013373      000      .BYTE      0
2049 013374      000      .BYTE      0
2050 013375      000      .BYTE      0
2051 013376      000      .BYTE      0
2052 013377      000      .BYTE      0
2053 013400      000      .BYTE      0
2054 013401      000      .BYTE      0
2055 013402      000      .BYTE      0
2056 013403      000      .BYTE      0
2057 013404      000      .BYTE      0
2058 013405      000      .BYTE      0
2059 013406      000      .BYTE      0
2060 013407      000      .BYTE      0
2061 013410      000      .BYTE      0
2062 013411      000      .BYTE      0
2063 013412      000      .BYTE      0
2064 013413      000      .BYTE      0
2065 013414      000      .BYTE      0
2066 013415      000      .BYTE      0
2067 013416      000      .BYTE      0
2068 013417      000      .BYTE      0
2069 013420      000      .BYTE      0
2070 013421      000      .BYTE      0
2071 013422      000      .BYTE      0
2072 013423      000      .BYTE      0
2073 013424      000      .BYTE      0
2074 013425      000      .BYTE      0
2075 013426      000      .BYTE      0
2076 013427      000      .BYTE      0
2077 013430      000      .BYTE      0
2078 013431      000      .BYTE      0
2079 013432      000000    .WORD      0
2080 013434      001001    REM.TBL: .WORD 1001
2081 013436      004004    .WORD     4004
2082 013440      020020    .WORD    20020
```


22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
2084      :ML4
2085      :
2086
2087 013442 003003      .WORD 3003
2088 013444 014014      .WORD 14014
2089 013446 021460      .WORD 21460
2090 013450 005005      .WORD 5005
2091 013452 024024      .WORD 24024
2092 013454 023023      .WORD 23023
2093 013456 017017      .WORD 17017
2094 013460 035474      .WORD 35474
2095 013462 024465      .WORD 24465
2096 013464 021021      .WORD 21021
2097 013466 007007      .WORD 7007
2098 013470 034034      .WORD 34034
2099 013472 022463      .WORD 22463
2100 013474 011011      .WORD 11011
2101 013476 005444      .WORD 5444
2102 013500 026026      .WORD 26026
2103 013502 033033      .WORD 33033
2104 013504 016457      .WORD 16457
2105 013506 033472      .WORD 33472
2106 013510 014455      .WORD 14455
2107 013512 023462      .WORD 23462
2108 013514 015015      .WORD 15015
2109 013516 025464      .WORD 25464
2110 013520 025025      .WORD 25025
2111 013522 027027      .WORD 27027
2112 013524 037037      .WORD 37037
2113 013526 036477      .WORD 36477
2114 013530 030471      .WORD 30471
2115 013532      041      .BYTE 41
2116      .EVEN
2117 013534 064477      DT.1: .WORD 64477
2118 013536 007700      .WORD 7700
2119 013540      000      .BYTE 0
2120 013541      000      .BYTE 0
2121 013542 064477      .WORD 64477
2122 013544 000077      .WORD 77
2123 013546      000      .BYTE 0
2124 013547      000      .BYTE 0
2125 013550 100577      .WORD -77201
2126 013552 017700      .WORD 17700
2127 013554      077      .BYTE 77
2128 013555      000      .BYTE 0
2129 013556 100577      .WORD -77201
2130 013560 000000      .WORD 0
2131 013562      077      .BYTE 77
2132 013563      000      .BYTE 0
2133 013564 100577      .WORD -77201
2134 013566 010077      .WORD 10077
2135 013570      077      .BYTE 77
2136      .EVEN
2137 013572      RH.ADD: .BLKW 1
2138 013574      RH.TYP: .BLKW 1
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

2140		:ML4	
2141		:	
2142			
2143	013576	RH.VEC: .BLKW	1
2144	013600	ML.LUN: .BLKW	1
2145	013602	ML.DUT: .BLKW	1
2146			
2147			
2148	100000	BIT15==	-100000
2149	040000	BIT14==	40000
2150	020000	BIT13==	20000
2151	010000	BIT12==	10000
2152	004000	BIT11==	4000
2153	002000	BIT10==	2000
2154	001000	BIT09==	1000
2155	000400	BIT08==	400
2156	000200	BIT07==	200
2157	000100	BIT06==	100
2158	000040	BIT05==	40
2159	000020	BIT04==	20
2160	000010	BIT03==	10
2161	000004	BIT02==	4
2162	000002	BIT01==	2
2163	000001	BIT00==	1
2164	001000	BIT9==	1000
2165	000400	BIT8==	400
2166	000200	BIT7==	200
2167	000100	BIT6==	100
2168	000040	BIT5==	40
2169	000020	BIT4==	20
2170	000010	BIT3==	10
2171	000004	BIT2==	4
2172	000002	BIT1==	2
2173	000001	BIT0==	1
2174	000040	EF.START==	40
2175	000037	EF.RESTART==	37
2176	000036	EF.CONTINUE==	36
2177	000035	EF.NEW==	35
2178	000034	EF.PWR==	34
2179	000340	PRI07==	340
2180	000300	PRI06==	300
2181	000240	PRI05==	240
2182	000200	PRI04==	200
2183	000140	PRI03==	140
2184	000100	PRI02==	100
2185	000040	PRI01==	40
2186	000000	PRI00==	0
2187	000004	EVL==	4
2188	000010	LOT==	10
2189	000020	ADR==	20
2190	000040	IDU==	40
2191	000100	ISR==	100
2192	000200	UAM==	200
2193	000400	BOE==	400
2194	001000	PNT==	1000

2196		:ML4	
2197		:	
2198		:	
2199	002000	PRI==	2000
2200	004000	IXE==	4000
2201	010000	IBE==	10000
2202	020000	IER==	20000
2203	040000	LOE==	40000
2204	100000	HOE==	-100000
2205	004222	FMT.1=	P.AAA
2206	004266	FMT.2=	P.AAB
2207	004354	FMT.3=	P.AAC
2208	004400	FMT.4=	P.AAD
2209	004430	FMT.5=	P.AAE
2210	004532	FMT.6=	P.AAF
2211	004562	FMT.7=	P.AAG
2212	004610	FMT.8=	P.AAH
2213	004644	FMT.9=	P.AAI
2214	004676	FMT.10=	P.AAJ
2215	004750	FMT.11=	P.AAK
2216	005004	FMT.12=	P.AAL
2217	005034	FMT.13=	P.AAM
2218	005110	FMT.14=	P.AAN
2219	005150	FMT.15=	P.AAO
2220	005216	FMT.16=	P.AAP
2221	005306	FMT.17=	P.AAQ
2222	005342	FMT.18=	P.AAR
2223	005412	FMT.19=	P.AAS
2224	005462	FMT.20=	P.AAT
2225	005534	FMT.21=	P.AAU
2226	005606	FMT.22=	P.AAV
2227	005652	FMT.23=	P.AAW
2228	005722	FMT.24=	P.AAX
2229	005746	FMT.25=	P.AAY
2230	005764	FMT.26=	P.AAZ
2231	006016	ONE.FMT=	P.ABA
2232	006024	TWO.FMT=	P.ABB
2233	006034	THR.FMT=	P.ABC
2234	006046	FOR.FMT=	P.ABD
2235	006062	FIV.FMT=	P.ABE
2236	006100	SIX.FMT=	P.ABF
2237	006120	SEV.FMT=	P.ABG
2238	006142	EIG.FMT=	P.ABH
2239	006166	NIN.FMT=	P.ABI
2240	006214	TEN.FMT=	P.ABJ
2241	006244	ELV.FMT=	P.ABK
2242	006276	WRD.1=	P.ABL
2243	006302	WRD.2=	P.ABM
2244	006314	WRD.3=	P.ABN
2245	006322	WRD.4=	P.ABO
2246	006330	WRD.5=	P.ABP
2247	006336	WRD.6=	P.ABQ
2248	006344	WRD.7=	P.ABR
2249	006360	WRD.8=	P.ABS
2250	006374	WRD.9=	P.ABT

2252		:ML4	
2253		:	
2254			
2255	006406	WRD.10=	P.ABU
2256	006416	WRD.11=	P.ABV
2257	006426	WRD.12=	P.ABW
2258	006436	WRD.13=	P.ABX
2259	006442	WRD.14=	P.ABY
2260	006454	WRD.15=	P.ABZ
2261	006462	WRD.16=	P.ACA
2262	006470	WRD.17=	P.ACB
2263	006502	WRD.18=	P.ACC
2264	006506	WRD.19=	P.ACD
2265	006514	WRD.20=	P.ACE
2266	006522	WRD.21=	P.ACF
2267	006530	WRD.22=	P.ACG
2268	006544	WRD.23=	P.ACH
2269	006552	WRD.24=	P.ACI
2270	006560	WRD.25=	P.ACJ
2271	006574	WRD.26=	P.ACK
2272	006602	WRD.27=	P.ACL
2273	006610	WRD.29=	P.ACM
2274	006624	WRD.30=	P.ACN
2275	006632	WRD.31=	P.ACO
2276	006646	WRD.32=	P.ACP
2277	006654	WRD.33=	P.ACQ
2278	006660	WRD.34=	P.ACR
2279	006666	WRD.35=	P.ACS
2280	006674	WRD.36=	P.ACT
2281	006700	WRD.37=	P.ACU
2282	006710	WRD.38=	P.ACV
2283	006716	WRD.39=	P.ACW
2284	006726	WRD.40=	P.ACX
2285	006732	WRD.41=	P.ACY
2286	006746	WRD.42=	P.ACZ
2287	006754	WRD.43=	P.ADA
2288	006764	WRD.44=	P.ADB
2289	006772	WRD.45=	P.ADC
2290	007004	WRD.46=	P.ADD
2291	007016	WRD.47=	P.ADE
2292	007026	WRD.48=	P.ADF
2293	007036	WRD.49=	P.ADG
2294	007046	WRD.50=	P.ADH
2295	007054	WRD.51=	P.ADI
2296	007066	WRD.52=	P.ADJ
2297	007074	WRD.53=	P.ADK
2298	007104	WRD.54=	P.ADL
2299	007114	WRD.55=	P.ADM
2300	007122	WRD.56=	P.ADN
2301	007130	WRD.57=	P.ADO
2302	007140	WRD.58=	P.ADP
2303	007152	WRD.59=	P.ADQ
2304	007156	WRD.60=	P.ADR
2305	007174	WRD.61=	P.ADS
2306	007210	WRD.62=	P.ADT

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

2308		:ML4	
2309		:	
2310			
2311	007216	WRD.63=	P.ADU
2312	007224	WRD.64=	P.ADV
2313	007232	WRD.65=	P.ADW
2314	007240	WRD.67=	P.ADX
2315	007246	WRD.68=	P.ADY
2316	007252	WRD.69=	P.ADZ
2317	007260	WRD.70=	P.AEA
2318	007272	WRD.71=	P.AEB
2319	007302	WRD.72=	P.AEC
2320	007310	WRD.73=	P. D
2321	007316	WRD.74=	P.AEE
2322	007330	WRD.75=	P.AEF
2323	007340	WRD.76=	P.AEG
2324	007346	WRD.77=	P.AEH
2325	007366	PHR.1=	P.AEI
2326	007404	PHR.2=	P.AEJ
2327	007422	PHR.3=	P.AEK
2328	007454	PHR.4=	P.AEL
2329	007472	PHR.5=	P.AEM
2330	007504	PHR.6=	P.AEN
2331	007516	PHR.7=	P.AEO
2332	007540	PHR.8=	P.AEP
2333	007552	PHR.9=	P.AEQ
2334	007564	PHR.10=	P.AER
2335	007576	PHR.11=	P.AES
2336	007614	PHR.12=	P.AET
2337	007636	PHR.13=	P.AEU
2338	007662	PHR.14=	P.AEV
2339	007714	PHR.15=	P.AEW
2340	007726	FNC.1=	P.AEX
2341	007742	FNC.2=	P.AEY
2342	007750	FNC.3=	P.AEZ
2343	007756	FNC.4=	P.AFA
2344	007774	FNC.5=	P.AFB
2345	010004	FNC.6=	P.AFC
2346	010012	FNC.7=	P.AFD
2347	010022	FNC.8=	P.AFE
2348	010036	FNC.9=	P.AFF
2349	010050	FNC.10=	P.AFG
2350	010060	FNC.11=	P.AFH
2351	010100	FNC.12=	P.AFI
2352	010112	FNC.13=	P.AFJ
2353	010122	FNC.14=	P.AFK
2354	010136	FNC.15=	P.AFL
2355	010150	FNC.16=	P.AFM
2356	010162	FNC.17=	P.AFN
2357	010172	FNC.18=	P.AFO
2358	010204	FNC.19=	P.AFP
2359	010216	FNC.21=	P.AFQ
2360	010230	FNC.22=	P.AFR
2361	010242	FNC.23=	P.AFS
2362	010256	REG.1=	P.AFT

```

2364      :ML4
2365      :
2366
2367      010266      REG.2=      P.AFU
2368      010274      REG.3=      P.AFV
2369      010302      REG.4=      P.AFW
2370      010310      REG.5=      P.AFX
2371      010316      REG.6=      P.AFY
2372      010324      REG.7=      P.AFZ
2373      010332      REG.8=      P.AGA
2374      010340      REG.9=      P.AGB
2375      010346      REG.10=     P.AGC
2376      010354      REG.11=     P.AGD
2377      010362      REG.12=     P.AGE
2378      010370      REG.13=     P.AGF
2379      010376      REG.14=     P.AGG
2380      010404      REG.15=     P.AGH
2381      010412      REG.16=     P.AGI
2382      010420      REG.17=     P.AGJ
2383      010430      REG.18=     P.AGK
2384      010436      REG.19=     P.AGL
2385      010444      REG.20=     P.AGM
2386      010454      REG.21=     P.AGN
2387      010464      ASYNC=      P.AGO
2388      010526      SYNC=      P.AGP
2389      010570      ARR.DAT=   P.AGQ
2390      010630      MEM.ARR=   P.AGR
2391      010672      INTER=    P.AGS
2392      010732      DATA.LATE= P.AGT
2393      010772      SC.SET=    P.AGU
2394      011026      TRBLE.LOOP= P.AGV
2395      011060      RH.ERROR=  P.AGW
2396      011106      TIME.OUT=  P.AGX
2397      011152      D1.TEMP=   NIB.SAVE
2398      011154      D2.TEMP=   NIB.SAVE+2
2399      011156      E2.TEMP=   NIB.SAVE+4
2400
2401
2402

```

```

2406 013604
2407 013604 004167 170250      LOAD.STACK:
2408 013610 016601 000012      JSR      R1,$SAVE3
2409 013614 012702 012600      MOV      12(SP),R1      ; NIB.PTR,*
2410 013620 060102      MOV      #STK.OFF,R2
2411 013622 016601 000014      ADD      R1,R2
2412 013626 016600 000012      MOV      14(SP),R1      ; STK.PTR,*
2413 013632 006300      MOV      12(SP),R0      ; NIB.PTR,*
2414 013634 066007 013640      ASL      R0
2415 013640 000024      ADD      1$(R0),PC
2416 013642 000042      1$:      .WORD  2$-1$
           .WORD  3$-1$

```

```

2418          :ML4
2419          :
2420
2421 013644 000060      .WORD 4$-1$
2422 013646 000076      .WORD 5$-1$
2423 013650 000114      .WORD 6$-1$
2424 013652 000132      .WORD 7$-1$
2425 013654 000160      .WORD 9$-1$
2426 013656 000176      .WORD 10$-1$
2427 013660 000224      .WORD 12$-1$
2428 013662 000260      .WORD 15$-1$
2429 013664 005003      2$: CLR R3 ; 2272
2430 013666 151203      BISB (R?),R3
2431 013670 010100      MOV R1,R0
2432 013672 160300      SUB R3,R0
2433 013674 016703 175252  MOV NIB.SAVE,R3
2434 013700 000500      BR 14$
2435 013702 005003      3$: CLR R3 ; 2276
2436 013704 151203      BISB (R2),R3
2437 013706 010100      MOV R1,R0
2438 013710 160300      SUB R3,R0
2439 013712 016703 175234  MOV NIB.SAVE,R3
2440 013716 000433      BR 8$
2441 013720 005003      4$: CLR R3 ; 2280
2442 013722 151203      BISB (R2),R3
2443 013724 010100      MOV R1,R0
2444 013726 160300      SUB R3,R0
2445 013730 016703 175216  MOV NIB.SAVE,R3
2446 013734 000461      BR 13$
2447 013736 005003      5$: CLR R3 ; 2284
2448 013740 151203      BISB (R2),R3
2449 013742 010100      MOV R1,R0
2450 013744 160300      SUB R3,R0
2451 013746 016703 175200  MOV NIB.SAVE,R3
2452 013752 000437      BR 11$
2453 013754 005003      6$: CLR R3 ; 2288
2454 013756 151203      BISB (R2),R3
2455 013760 010100      MOV R1,R0
2456 013762 160300      SUB R3,R0
2457 013764 016703 175164  MOV NIB.SAVE+2,R3
2458 013770 000444      BR 14$
2459 013772 005003      7$: CLR R3 ; 2292
2460 013774 151203      BISB (R2),R3
2461 013776 010100      MOV R1,R0
2462 014000 160300      SUB R3,R0
2463 014002 016703 175146  MOV NIB.SAVE+2,R3
2464 014006 006203      8$: ASR R3
2465 014010 006203      ASR R3
2466 014012 006203      ASR R3
2467 014014 006203      ASR R3
2468 014016 000431      BR 14$
2469 014020 005003      9$: CLR R3 ; 2296
2470 014022 151203      BISB (R2),R3
2471 014024 010100      MOV R1,R0
2472 014026 160300      SUB R3,R0

```

```

2474      ;ML4
2475      ;
2476
2477 014030 016703 175120      MOV      NIB.SAVE+2,R3
2478 014034 000421            BR        13$
2479 014036 005003      10$:  CLR      R3                      ;
2480 014040 151203            BISB     (R2),R3
2481 014042 010100            MOV      R1,R0
2482 014044 160300            SUB      R3,R0
2483 014046 016703 175102      MOV      NIB.SAVE+2,R3
2484 014052 006203      11$:  ASR      R3
2485 014054 006203            ASR      R3
2486 014056 006203            ASR      R3
2487 014060 006203            ASR      R3
2488 014062 000406            BR        13$
2489 014064 005003      12$:  CLR      R3                      ;
2490 014066 151203            BISB     (R2),R3
2491 014070 010100            MOV      R1,R0
2492 014072 160300            SUB      R3,R0
2493 014074 016703 175056      MOV      NIB.SAVE+4,R3
2494 014100 000303      13$:  SWAB     R3
2495 014102 042703 177760      14$:  BIC      #177760,R3
2496 014106 105060 012612      CLRB     STACK(R0)
2497 014112 150360 012612      BISB     R3,STACK(R0)
2498 014116 000207            RTS      PC                      ;
2499 014120 005003      15$:  CLR      R3                      ;
2500 014122 151203            BISB     (R2),R3
2501 014124 160301            SUB      R3,R1
2502 014126 016703 175024      MOV      NIB.SAVE+4,R3
2503 014132 006203            ASR      R3
2504 014134 006203            ASR      R3
2505 014136 006203            ASR      R3
2506 014140 006203            ASR      R3
2507 014142 000303            SWAB     R3
2508 014144 042703 177770      BIC      #177770,R3
2509 014150 105061 012612      CLRB     STACK(R1)
2510 014154 150361 012612      BISB     R3,STACK(R1)
2511 014160 000207            RTS      PC                      ;
2512
2513      ; Routine Size: 119 words
2514      ; Maximum stack depth per invocation: 4 words
2519
2520

```

2301

2304

2268
2308

2217

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (6)

2522 :ML4
2523 :
2524 :
2525 :
2526 :
2527 :
2528 :
2529 :
2530 :
2531 :
2532 :
2533 :
2534 :
2535 :
2536 :
2537 :
2538 :
2539 :
2540 :
2541 :
2542 :
2543 :
2544 :
2545 :
2546 :
2547 :
2548 :
2552 :
2556 014162
2557 014162
2558 014170
2559 014176
2560 014204
2561 014212
2562 :
2563 :
2564 :
2569 :
2570 :

2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336

routine DAT_DM_XFER : novalue =

++

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 COMPIMENT 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

2332
2333
2334
2335
2313

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

2572 :ML4
2573 :
2574 :
2575 :
2576 :
2577 :
2578 :
2579 :
2580 :
2581 :
2582 :
2583 :
2584 :
2585 :
2586 :
2587 :
2588 :
2589 :
2590 :
2591 :
2592 :
2593 :
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 :

```

2337 routine STRIPPER (WRD_CNT, NIB_CNT) : novalue =
2338   begin
2339
2340   !++
2341   FUNCTIONAL DESCRIPTION:
2342   STRIPPER RUNS IN DATA DIAG MODE AND STRIPS OUT
2343   A VARIABLE NUMBER OF GOOD NIBBLES FROM THE
2344   GOOD BLOCK AND STORES THEM INTO A CONTIGIOUS
2345   STACK FOR SEQUENTIAL ACCESSING
2346
2347   FORMAL PARAMETERS:
2348   WRD_CNT
2349   TEL[S STRIPPER HOW MANY WORDS IN THE GOOD
2350   BLOCK TO READ OUT
2351
2352   NIB_CNT
2353   TEL[S STRIPPER HOW MANY NIBBLES TO
2354   STRIP OUT OF EACH WORD.
2355
2356   IMPLICIT INPUTS:
2357
2358
2359   local
2360     STK_PTR;
2361
2362   incr CNT from 0 to .NIB_CNT do
2363     STK_OFF [.CNT] = ZEROES;
2364
2365   CLR_MBUS;
2366   STK_PTR = -1;
2367   DAT_DM_XFER ();
2368   MLC51 = read;
2369   DELAY (ONE_US);
2370
2371   incr CNT from 0 to .WRD_CNT do
2372     begin
2373       PD_TEMP = .MLPD;
2374       DAT_CLK = ONE;
2375       DELAY (ONE_US);
2376       RD_LNG_WRD;
2377
2378       incr NIB_PTR from 0 to .NIB_CNT do
2379         begin
2380           STK_PTR = .STK_PTR + 1;
2381
2382           if .PD_TEMP [.NIB_PTR] IS_SET
2383             then
2384               STK_OFF [.NIB_PTR] = (.STK_OFF [.NIB_PTR]) + (.NIB_CNT + 1)
2385             !THEN INCREMENT IT'S STACK OFFSET
2386           else
2387             LOAD_STACK (.STK_PTR, .NIB_PTR);
2388

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (7)

```

!STACK POINTER
!CLEAR OUT THE STACK OFFSETS
!RESET THE STACK POINTER
!SET UP A DATA XFER AT THE GOOD BLOCK
!DO A READ XFER
!ALLOW PROM DATA TO GET INTO THE MLPD REG
!LOAD THE STACK WITH ALL GOOD NIBBLE DATA
!GET THE PROM DATA
!CLOCK OUT THE DATA WORD
!READ DATA DIAG REGS INTO NIBBLE SAVE
!STRIP OUT X NUMBER OF NIBBLES
!INCREMENT THE STACK POINTER
!SEE IF THIS A GOOD NIBBLE
!ELSE LOAD THE NIBBLE IN THE STACK

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (7)

```

2628 :ML4
2629 :
2630 :
2631 :      2389      end;
2632 :      2390
2633 :      2391      end;
2634 :      2392
2635 :      2393      CLR_MBUS;
2636 :      2394      end;
2640 :
2641 :
2642 :
2643 :
2647 014214      STRIPPER:
2648 014214 004167 167676      JSR      R1,$$SAVES      ;      2337
2649 014220 005746      TST      -(SP)      ;
2650 014222 016604 000020      MOV      20(SP),R4      ; NIB.CNT,*      2362
2651 014226 005001      CLR      R1      ; CNT
2652 014230 000403      BR      2$
2653 014232 105061 012600      1$:      CLRB     STK.OFF(R1)      ; *(CNT)      2363
2654 014236 005201      INC      R1      ; CNT      2362
2655 014240 020104      2$:      CMP      R1,R4      ; CNT,*
2656 014242 003773      BLE     1$
2657 014244 152777 000040 176742      BISB     #40,@ML.REG+40      ;      2363
2658 014252 016705 177324      MOV      ML,DUT,R5
2659 014256 042705 177770      BIC      #177770,R5
2660 014262 142777 000007 176724      BICB     #7,@ML.REG+40
2661 014270 150577 176720      BISB     R5,@ML.REG+40
2662 014274 012705 177777      MOV      #-1,P5      ; *,STK.PTR      2366
2663 014300 004767 177656      JSR      PC,DAT.DM.XFER      ;      2367
2664 014304 012777 000071 176642      MOV      #71,@ML.REG      ;      2368
2665 014312 012701 000001      MOV      #1,R1      ; *,SSTMP2      2369
2666 014316 001410      3$:      BEQ      6$
2667 014320 016702 165572      MOV      LSDLY,R2      ; *,SSTMP1
2668 014324 001403      BEQ      5$
2669 014326 005016      4$:      CLR      (SP)      ; SSTMP
2670 014330 005302      DEC      R2      ; SSTMP1
2671 014332 001375      BNE     4$
2672 014334 005301      5$:      DEC      R1      ; SSTMP2
2673 014336 000767      BR      3$
2674 014340 005003      6$:      CLR      R3      ; CNT      2371
2675 014342 000502      BR      16$
2676 014344 017767 177034 176546      7$:      MOV      @ML.REG+230,PD.TEMP      ;      2373
2677 014352 152777 000020 176714      BISB     #20,@ML.REG+120      ;      2374
2678 014360 012701 000001      MOV      #1,R1      ; *,SSTMP2      2375
2679 014364 001410      8$:      BEQ      11$
2680 014366 016702 165524      MOV      LSDLY,R2      ; *,SSTMP1
2681 014372 001403      BEQ      10$

```

```

2683      ;ML4
2684      ;
2685
2686 014374 005016      9S:   CLR      (SP)      : $STMP
2687 014376 005302      DEC      R2         : $STMP1
2688 014400 001375      BNE      9S
2689 014402 005301      10S:  DEC      R1         : $STMP2
2690 014404 000767      BR       8S
2691 014406 017767 176732 174536 11S:  MOV      @ML.REG+170,D1.TEMP
2692 014414 017767 176734 174532      MOV      @ML.REG+200,D2.TEMP
2693 014422 017767 176706 174526      MOV      @ML.REG+160,E2.TEMP
2694 014430 005002      CLR      R2         : NIB.PTR      2378
2695 014432 000443      BR       15S
2696 014434 005205      12S:  INC      R5         : STK.PTR      2380
2697 014436 010201      MOV      R2,R1      : NIB.PTR,*    2382
2698 014440 006201      ASR      R1
2699 014442 006201      ASR      R1
2700 014444 006201      ASR      R1
2701 014446 062701 013120      ADD      #PD.TEMP,R1
2702 014452 010146      MOV      R1,-(SP)
2703 014454 010246      MOV      R2,-(SP)      : NIB.PTR,*
2704 014456 042716 177770      BIC      #177770,(SP)
2705 014462 012746 000001      MOV      #1,-(SP)
2706 014466 005046      CLR      -(SP)
2707 014470 004767 166444      JSR      PC,BL$GT2
2708 014474 062706 000010      ADD      #10,SP
2709 014500 005300      DEC      R0
2710 014502 001011      BNE      13S
2711 014504 005001      CLR      R1
2712 014506 156201 012600      BISB     STK.OFF(R2),R1  : *(NIB.PTR),*  2384
2713 014512 060401      ADD      R4,R1
2714 014514 010100      MOV      R1,R0
2715 014516 005200      INC      R0
2716 014520 110062 012600      MOV      R0,STK.OFF(R2)  : *,*(NIB.PTR)
2717 014524 000405      BR       14S
2718 014526 010546      13S:  MOV      R5,-(SP)      : STK.PTR,*    2382
2719 014530 010246      MOV      R2,-(SP)      : NIB.PTR,*    2387
2720 014532 004767 177046      JSR      PC,LOAD.STACK
2721 014536 022626      CMP      (SP)+,(SP)+
2722 014540 005202      14S:  INC      R2         : NIB.PTR      2378
2723 014542 020204      15S:  CMP      R2,R4       : NIB.PTR,*
2724 014544 003733      BLE      12S
2725 014546 005203      INC      R3
2726 014550 020366 000022      16S:  CMP      R3,22(SP)     : CNT          2371
2727 014554 003673      BLE      7S
2728 014556 152777 000040 176430      BISB     #40,@ML.REG+40
2729 014564 016705 177012      MOV      ML,DUT,R5
2730 014570 042705 177770      BIC      #177770,R5
2731 014574 142777 000007 176412      BICB     #7,@ML.REG+40
2732 014602 150577 176406      BISB     R5,@ML.REG+40
2733 014606 005726      TST      (SP)+
2734 014610 000207      RTS      PC
2735
2736      : Routine Size: 127 words
2737      : Maximum stack depth per invocation: 11 words

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 95^{E 7}

SEQ 0082

2739
2740
2741
2746
2747

:ML4
:

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (8)

```

2749 :ML4
2750 :
2751 :
2752 : 2395 routine CAL_CRC : novalue =
2753 : 2396   begin
2754 : 2397
2755 : 2398   ++
2756 : 2399   FUNCTIONAL DESCRIPTION:
2757 : 2400     CALCULATE THE CRC CODE FOR THE GOOD BLOCKS
2758 : 2401     FIRST CRC GROUP (60 NIBBLES) BY STRIPPING
2759 : 2402     OUT GOOD NIBBLES AND MODULO 2 ADDING EACH
2760 : 2403     BITS WEIGHT RESULTING IN A_CAL,B_CAL,P_CAL.
2761 : 2404
2762 : 2405   IMPLICIT INPUTS:
2763 : 2406     P_CAL,A_CAL,B_CAL
2764 : 2407     GLOBAL STORAGE LOCATION WHERE CALCULATED
2765 : 2408     CRC CODES ARE STORED AND EXAMINED FROM
2766 : 2409
2767 : 2410     REM TBL
2768 : 2411     TABLE CONTAINING PRECALCULATED BIT POSITION
2769 : 2412     WEIGHTS USED IN THE MODULO 2 CALCULATION OF THE
2770 : 2413     CRC CODE
2771 : 2414   --
2772 : 2415
2773 : 2416   local
2774 : 2417     CHANNEL,           !CHANNEL POINTS TO THE BITS IN A WORD
2775 : 2418     STK_PTR,         !STACK POINTER
2776 : 2419     NIB_SAV : bitvector [4], !NIBBLE SAVE LOCATION
2777 : 2420     ALOG,           !INDEX INTO REMAINDER TABLE
2778 : 2421     BLOG;         !INDEX INTO REMAINDER TABLE
2779 : 2422
2780 : 2423   STRIPPER (12, 9);
2781 : 2424   STK_PTR = -1;
2782 : 2425
2783 : 2426   incr PLOG from 0 to 5 do
2784 : 2427     begin
2785 : 2428     CHANNEL = -1;
2786 : 2429
2787 : 2430     incr NIB_CNT from 0 to 8 do
2788 : 2431     begin
2789 : 2432     STK_PTR = .STK_PTR + 1;
2790 : 2433     NIB_SAV = .stack [.STK_PTR];
2791 : 2434
2792 : 2435     incr BIT_TST from 0 to 3 do
2793 : 2436     begin
2794 : 2437     CHANNEL = .CHANNEL + 1;
2795 : 2438
2796 : 2439     if .NIB_SAV [.BIT_TST] IS_SET
2797 : 2440     then
2798 : 2441     begin
2799 : 2442     ALOG = .PLOG + .CHANNEL;
2800 : 2443     BLOG = .PLOG + .CHANNEL*2;
2801 : 2444
2802 : 2445     while .ALOG geq 63 do
2803 : 2446     ALOG = .ALOG - 63;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (8)

```

2805 ;ML4
2806 :
2807 :
2808 : 2447
2809 : 2448 while .BLOG geq 63 do !REDUCE BLOG UNTIL <64
2810 : 2449 BLOG = .BLOG - 63;
2811 : 2450
2812 : 2451 P_CAL = (.P_CAL) xor (.REM_TBL [.PLOG]); !MOD 2 P_CAL WITH REM_TBL
2813 : 2452 A_CAL = (.A_CAL) xor (.REM_TBL [.ALOG]); !MOD 2 A_CAL WITH REM_TBL
2814 : 2453 B_CAL = (.B_CAL) xor (.REM_TBL [.BLOG]); !MOD 2 B_CAL WITH REM_TBL
2815 : 2454 end;
2816 : 2455
2817 : 2456 end;
2818 : 2457
2819 : 2458 end;
2820 : 2459
2821 : 2460 STK_PTR = .STK_PTR + 1; !SKIP OVER THE CRC NIB NT THE STACK
2822 : 2461 end;
2823 : 2462
2824 : 2463 end;
2828 :
2832 014612 004167 167300 CAL.CRC:JSR R1,$SAVE5 ; 2395
2833 014616 162706 000012 SUB #12,SP ;
2834 014622 012746 000014 MOV #14,-(SP) ; 2423
2835 014626 012746 000011 MOV #11,-(SP) ;
2836 014632 004767 177356 JSR PC,STRIPPER
2837 014636 012766 177777 000006 MOV #-1,6(SP) ; *,STK.PTR 2424
2838 014644 005066 000004 CLR 4(SP) ; PLOG 2426
2839 014650 012766 177777 000010 1$: MOV #-1,10(SP) ; *,CHANNEL 2428
2840 014656 005066 000012 CLR 12(SP) ; NIB.CNT 2430
2841 014662 005266 000006 2$: INC 6(SP) ; STK.PTR 2432
2842 014666 016605 000006 MOV 6(SP),R5 ; STK.PTR,* 2433
2843 014672 116566 012612 000014 MOVB STACK(R5),14(SP) ; *,NIB.SAV
2844 014700 105066 000015 CLRB 15(SP) ; NIB.SAV
2845 014704 005003 CLR R3 ; BIT.TST 2435
2846 014706 005266 000010 3$: INC 10(SP) ; CHANNEL 2437
2847 014712 010305 MOV R3,R5 ; BIT.TST,* 2439
2848 014714 006205 ASR R5
2849 014716 006205 ASR R5
2850 014720 006205 ASR R5
2851 014722 012704 000014 MOV #14,R4
2852 014726 060604 ADD SP,R4 ; NIB.SAV,*
2853 014730 060405 ADD R4,R5
2854 014732 010546 MOV R5,-(SP)
2855 014734 010346 MOV R3,-(SP) ; BIT.TST,*
2856 014736 042716 177770 BIC #177770,(SP)
2857 014742 012746 000001 MOV #1,-(SP)
2858 014746 005046 CLR -(SP)

```

2860									
2861									
2862									
2863	014750	004767	166164		JSR	PC,BLSGT2			
2864	014754	062706	000010		ADD	#10,SP			
2865	014760	005300			DEC	R0			
2866	014762	001066			BNE	7\$			
2867	014764	016601	000004		MOV	4(SP),R1	:	PLOG,ALOG	2442
2868	014770	066601	000010		ADD	10(SP),R1	:	CHANNEL,ALOG	
2869	014774	016605	000010		MOV	10(SP),R5	:	CHANNEL,*	2443
2870	015000	006305			ASL	R5			
2871	015002	066605	000004		ADD	4(SP),R5	:	PLOG,*	
2872	015006	010502			MOV	R5,R2	:	*,BLOG	
2873	015010	020127	000077	4\$:	CMP	R1,#77	:	ALOG,*	2445
2874	015014	002403			BLT	5\$			
2875	015016	162701	000077		SUB	#77,R1	:	*,ALOG	2446
2876	015022	000772			BR	4\$:		2445
2877	015024	020227	000077	5\$:	CMP	R2,#77	:	BLOG,*	2448
2878	015030	002403			BLT	6\$			
2879	015032	162702	000077		SUB	#77,R2	:	*,BLOG	2449
2880	015036	000772			BR	5\$:		2448
2881	015040	005005		6\$:	CLR	R5	:		2451
2882	015042	016604	000004		MOV	4(SP),R4	:	PLOG,*	
2883	015046	156405	013434		BISB	REM.TBL(R4),R5			
2884	015052	010546			MOV	R5,-(SP)			
2885	015054	046716	176064		BIC	P.CAL,(SP)			
2886	015060	040567	176060		BIC	R5,P.CAL			
2887	015064	052667	176054		BIS	(SP)+,P.CAL			
2888	015070	005005			CLR	R5	:		2452
2889	015072	156105	013434		BISB	REM.TBL(R1),R5	:	*(ALOG),*	
2890	015076	010504			MOV	R5,R4			
2891	015100	046704	176034		BIC	A.CAL,R4			
2892	015104	040567	176030		BIC	R5,A.CAL			
2893	015110	050467	176024		BIS	R4,A.CAL			
2894	015114	005005			CLR	R5	:		2453
2895	015116	156205	013434		BISB	REM.TBL(R2),R5	:	*(BLOG),*	
2896	015122	010504			MOV	R5,R4			
2897	015124	046704	176012		BIC	B.CAL,R4			
2898	015130	040567	176006		BIC	R5,B.CAL			
2899	015134	050467	176002		BIS	R4,B.CAL			
2900	015140	005203		7\$:	INC	R3	:	BIT.TST	2435
2901	015142	020327	000003		CMP	R3,#3	:	BIT.TST,*	
2902	015146	003657			BLE	3\$			
2903	015150	005266	000012		INC	12(SP)	:	NIB.CNT	2430
2904	015154	026627	000012	000010	CMP	12(SP),#10	:	NIB.CNT,*	
2905	015162	003637			BLE	2\$			
2906	015164	005266	000006		INC	6(SP)	:	STK.PTR	2460
2907	015170	005266	000004		INC	4(SP)	:	PLOG	2426
2908	015174	026627	000004	000005	CMP	4(SP),#5	:	PLOG,*	
2909	015202	003622			BLE	1\$			
2910	015204	062706	000016		ADD	#16,SP	:		2395
2911	015210	000207			RTS	PC			
2912									
2913									
2914									

: Routine Size: 128 words
: Maximum stack depth per invocation: 17 words

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 99⁷

SEQ 0086

2916
2917
2918
2923
2924

:ML4
:

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BLJML4.BLI.2 (9)

```

2926 :ML4
2927 :
2928 :
2929 : 2464 routine ERR_CHK_CRC =
2930 : 2465     begin
2931 : 2466
2932 : 2467     !++
2933 : 2468     FUNCTIONAL DESCRIPTION:
2934 : 2469     'ERROR CHECK CRC' COMPARES HARDWARE GENERATED CRC
2935 : 2470     CODES TO SOFTWARE CALCULATED CRC CODE FOR ONE
2936 : 2471     CRC GROUP AND RETURNS A ONE ON DETECTION OF ERRORS
2937 : 2472
2938 : 2473     IMPLICIT INPUTS:
2939 : 2474     P_CAL,A_CAL,B_CAL
2940 : 2475     GLOBAL STORAGE LOCATIONS WHERE CALCULATED
2941 : 2476     CRC CODES ARE STORED AND EXAMINED FROM
2942 : 2477
2943 : 2478     P_GEN,A_GEN,B_GEN
2944 : 2479     GLOBAL STORAGE LOCATIONS WHERE HARDWARE GENERATED
2945 : 2480     CRC CODES ARE STORED AND EXAMINED FROM
2946 : 2481     --
2947 : 2482
2948 : 2483     local
2949 : 2484     POS,
2950 : 2485     TEMP;
2951 : 2486
2952 : 2487     POS = -1;
2953 : 2488
2954 : 2489     incr STK_PTR from 9 to 59 by 10 do
2955 : 2490     begin
2956 : 2491     POS = .POS + 1;
2957 : 2492     TEMP = .stack [.STK_PTR];
2958 : 2493     (P_GEN)<.POS, 1> = .TEMP<0, 1>;
2959 : 2494     (A_GEN)<.POS, 1> = not .TEMP<1, 1>;
2960 : 2495     (B_GEN)<.POS, 1> = not .TEMP<2, 1>;
2961 : 2496     end;
2962 : 2497
2963 : 2498     if (.P_GEN neq .P_CAL) or (.A_GEN neq .A_CAL) or (.B_GEN neq .B_CAL) then return ONE else return ZERO;
2964 : 2499
2965 : 2500     !COMPARE GENERATED CRC CODE TO CALCULATED CRC CODE
2966 : 2501
2967 :
2968 :
2969 :
2970 :
2974 015212 ERR_CHK_CRC:
2975 015212 004167 166642 JSR R1,$SAVE3
2976 015216 012701 177777 MOV #-1,R1
2977 015222 012702 000011 MOV #11,R2
2978 015226 005201 1$: INC R1
2979 015230 005003 CLR R3

```

2464
2487
2489
2491
2492

```

2981      ;ML4
2982      ;
2983
2984 015232 156203 012612      BISB   STACK(R2),R3      ; *(STK.PTR),TEMP
2985 015236 012746 013152      MOV    #P.GEN,-(SP)   ;
2986 015242 010146              MOV    R1,-(SP)       ; POS,*
2987 015244 012746 000001      MOV    #1,-(SP)       ;
2988 015250 010346              MOV    R3,-(SP)       ; TEMP,*
2989 015252 042716 177776      BIC    #177776,(SP)   ;
2990 015256 004767 166114      JSR    PC,BL$PU2     ;
2991 015262 012716 013146      MOV    #A.GEN,(SP)   ;
2992 015266 010146              MOV    R1,-(SP)       ; POS,*
2993 015270 012746 000001      MOV    #1,-(SP)       ;
2994 015274 005046              CLR    -(SP)          ;
2995 015276 032703 000002      BIT    #2,R3         ; *,TEMP
2996 015302 001401              BEQ    2$
2997 015304 005216              INC    (SP)
2998 015306 005116              2$:   COM    (SP)
2999 015310 004767 166062      JSR    PC,BL$PU2     ;
3000 015314 012716 013150      MOV    #B.GEN,(SP)   ;
3001 015320 010146              MOV    R1,-(SP)       ; POS,*
3002 015322 012746 000001      MOV    #1,-(SP)       ;
3003 015326 005046              CLR    -(SP)          ;
3004 015330 032703 000004      BIT    #4,R3         ; *,TEMP
3005 015334 001401              BEQ    3$
3006 015336 005216              INC    (SP)
3007 015340 005116              3$:   COM    (SP)
3008 015342 004767 166030      JSR    PC,BL$PU2     ;
3009 015346 062706 000024      ADD    #24,SP        ;
3010 015352 062702 000012      ADD    #12,R2        ; *,STK.PTR
3011 015356 020227 000073      CMP    R2,#73        ; STK.PTR,*
3012 015362 003721              BLE    1$
3013 015364 026767 175562 175552  CMP    P.GEN,P.CAL   ;
3014 015372 001010              BNE    4$
3015 015374 026767 175546 175536  CMP    A.GEN,A.CAL   ;
3016 015402 001004              BNE    4$
3017 015404 026767 175540 175530  CMP    B.GEN,B.CAL   ;
3018 015412 001403              BEQ    5$
3019 015414 012701 000001      4$:   MOV    #1,R1
3020 015420 000401              BR     6$
3021 015422 005001              5$:   CLR    R1
3022 015424 010100              6$:   MOV    R1,R0
3023 015426 000207              RTS    PC
3024
3025
3026
3031
3032
; Routine Size: 71 words
; Maximum stack depth per invocation: 14 words

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (10)

```

3034 :ML4
3035 :
3036 :
3037 : 2502 routine FIND_COMP_BIT (BITS_XFERED) =
3038 : 2503     begin
3039 : 2504
3040 : 2505     !++
3041 : 2506     FUNCTIONAL DESCRIPTION:
3042 : 2507     'FIND COMPLIMENT' BIT IS CALLED BY THE
3043 : 2508     SYNDROME DECODE TESTS AND INDEXES INTO
3044 : 2509     THE IO_BUF LOOKING FOR COMPLIMENTED
3045 : 2510     BITS. A COMPLIMENTED BIT IS BY DEF A ONE
3046 : 2511
3047 : 2512     A ONE IS RETURNED IF THE INDEXED BIT IS
3048 : 2513     NOT SET
3049 : 2514
3050 : 2515     FORMAL PARAMETERS:
3051 : 2516     BITS_XFERED
3052 : 2517     REPRESENTS THE BIT BEING TESTED
3053 : 2518     IN THE DRIVE AND FROM THIS THE
3054 : 2519     WORD AND BIT INDEX INTO THE
3055 : 2520     IO_BUF CAN BE CALCULATED
3056 : 2521
3057 : 2522     IMPLICIT INPUTS:
3058 : 2523     -
3059 : 2524
3060 : 2525     local
3061 : 2526     COMP_WRD,           !WORD WHERE COMPLIMENT BIT IS LOCATED
3062 : 2527     COMP_BIT,         !BIT THAT IS COMPLIMENTED
3063 : 2528     BUF_SAV : bitvector [16]; !STORES THE WORD IN WHICH THE BIT IS COMPLIMENTED
3064 : 2529
3065 : 2530     COMP_WRD = .BITS_XFERED/16; !CALCULATE THE COMP WORD
3066 : 2531     COMP_BIT = .BITS_XFERED mod 16; !CALCULATE THE COMP BIT
3067 : 2532     BUF_SAV = .IO_BUF [.COMP_WRD]; !LOAD THE COMP WORD INTO BUF_SAV
3068 : 2533     IO_BUF [.COMP_WRD] = ZEROS; !ZERO THE COMP WORD
3069 : 2534
3070 : 2535     if .BUF_SAV [.COMP_BIT] IS_NOT_SET then return ZERO else return ONE;
3071 : 2536
3072 : 2537     !FIND COMPLIMENTED BIT AND RETURN ERROR STATUS
3073 : 2538     end;
3077 :

```

```

3081 015430          FIND.COMP.BIT:
3082 015430 004167 166410          JSR    R1,$SAVE2          ;          2502
3083 015434 005746          TST    -(SP)
3084 015436 016646 000012          MOV    12(SP),-(SP)          ; BITS.XFERED,* 2530
3085 015442 012746 000020          MOV    #20, -(SP)
3086 015446 004767 166300          JSR    PC,BLSDIV
3087 015452 010001          MOV    R0,R1          ; *,COMP.WRD

```

```

3089                                     :ML4
3090                                     :
3091                                     :
3092 015454 016616 000016                MOV    16(SP), (SP)                : BITS.XFERED,*
3093 015460 012746 000020                MOV    #20, -(SP)
3094 015464 004767 166274                JSR    PC,BLSMOD
3095 015470 010002                        MOV    R0,R2                        : *,COMP.BIT
3096 015472 010100                        MOV    R1,R0                        : COMP.WRD,*
3097 015474 006300                        ASL    R0
3098 015476 016066 011600 000006        MOV    IO.BUF(R0),6(SP)            : *,BUF.SAV
3099 015504 005060 011600                CLR    IO.BUF(R0)
3100 015510 010200                        MOV    R2,R0                        : COMP.BIT,*
3101 015512 006200                        ASR    R0
3102 015514 006200                        ASR    R0
3103 015516 006200                        ASR    R0
3104 015520 012701 000006                MOV    #6,R1
3105 015524 060601                        ADD    SP,R1                        : BUF.SAV,*
3106 015526 060100                        ADD    R1,R0
3107 015530 010046                        MOV    R0, -(SP)
3108 015532 010246                        MOV    R2, -(SP)                    : COMP.BIT,*
3109 015534 042716 177770                BIC    #177770, (SP)
3110 015540 012746 000001                MOV    #1, -(SP)
3111 015544 005046                        CLR    -(SP)
3112 015546 004767 165366                JSR    PC,BLSGT2
3113 015552 062706 000010                ADD    #10,SP
3114 015556 005700                        TST    R0
3115 015560 001002                        BNE    1$
3116 015562 005001                        CLR    R1
3117 015564 000402                        BR     2$
3118 015566 012701 000001                1$: MOV    #1,R1
3119 015572 010100                2$: MOV    R1,R0
3120 015574 062706 000010                ADD    #10,SP
3121 015600 000207                RTS    PC
3122
3123                                     : Routine Size: 53 words
3124                                     : Maximum stack depth per invocation: 11 words
3129
3130

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

2531
2532
2533
2535
2503
2502

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (11)

```

3132 :ML4
3133 :
3134 :
3135 : 2539 routine FORCE_REM (PLOG, CHAN) : novalue =
3136 : 2540     begin
3137 : 2541
3138 : 2542     !++
3139 : 2543     FUNCTIONAL DESCRIPTION:
3140 : 2544     VIA ECC DM AND THE PRECALCULATED MODULO 2
3141 : 2545     REMAINDERS STORED IN REM_TBL[], THE
3142 : 2546     CRC_A, CRC_B AND PAR_CRC_WRD
3143 : 2547     ARE FORCED WITH ERRONEOUS CRC DATA TO
3144 : 2548     FORCE PREDICTABLE ECC ERRORS
3145 : 2549
3146 : 2550     FORMAL PARAMETERS:
3147 : 2551     PLOG
3148 : 2552     POINTS TO ONE OF THE SIX WORDS OF
3149 : 2553     A CRC GROUP
3150 : 2554
3151 : 2555     CHAN
3152 : 2556     POINTS TO ONE OF 36 CHANNELS IN
3153 : 2557     A CRC GROUP
3154 : 2558
3155 : 2559     IMPLICIT INPUTS:
3156 : 2560     --
3157 : 2561
3158 : 2562     local
3159 : 2563     ALOG,                !INDEX INTO REM_TBL
3160 : 2564     BLOG;              !INDEX INTO REM_TBL
3161 : 2565
3162 : 2566     ALOG = .PLOG + .CHAN;  !CALCULATE A_LOG
3163 : 2567     BLOG = .PLOG + .CHAN*2; !CALCULATE B_LOG
3164 : 2568
3165 : 2569     while .ALOG geq 63 do  !REDUCE A_LOG UNTIL < 64
3166 : 2570         ALOG = .ALOG - 63;
3167 : 2571
3168 : 2572     while .BLOG geq 63 do  !REDUCE B_LOG UNTIL < 64
3169 : 2573         BLOG = .BLOG - 63;
3170 : 2574
3171 : 2575     ECC_DM = ONE;         !SET ECC DIAG MODE
3172 : 2576     PAR_CRC_WRD = .REM_TBL [.PLOG]; !LOAD PAR_CRC_WRD WITH REM_TBL
3173 : 2577     CRC_A = .REM_TBL [.ALOG];      !LOAD CRC_A WITH REM_TBL
3174 : 2578     CRC_B = .REM_TBL [.BLOG];      !LOAD CRC_B WITH REM_TBL
3175 : 2579     end;
3179 :
3183 015602          FORCE.REM:
3184 015602 004167 166252          JSR    R1,$SAVE3
3185 015606 016600 000014          MOV    14(SP),R0
                                     :
                                     : PLOG,*

```

2539
2566

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

```

3187      ;ML4
3188      ;
3189
3190 015612 010002      MOV      R0,R2      ; *,ALOG
3191 015614 066602 000012  ADD      12(SP),R2  ; CHAN,ALOG
3192 015620 016601 000012  MOV      12(SP),R1  ; CHAN,*
3193 015624 006301      ASL      R1
3194 015626 060001      ADD      R0,R1
3195 015630 020227 000077      1$:    CMP      R2,#77    ; ALOG,*
3196 015634 002403      BLT      2$
3197 015636 162702 000077      SUB      #77,R2    ; *,ALOG
3198 015642 000772      BR       1$
3199 015644 020127 000077      2$:    CMP      R1,#77    ; BLOG,*
3200 015650 002403      BLT      3$
3201 015652 162701 000077      SUB      #77,R1    ; *,BLOG
3202 015656 000772      BR       2$
3203 015660 152777 000001 175406 3$:    BISB     #1,@ML.REG+120
3204 015666 116003 013434      MOVB     REM.TBL(R0),R3
3205 015672 000303      SWAB     R3
3206 015674 042703 140377      BIC      #140377,R3
3207 015700 042777 037400 175416      BIC      #37400,@ML.REG+150
3208 015706 050377 175412      BIS      R3,@ML.REG+150
3209 015712 116203 013434      MOVB     REM.TBL(R2),R3
3210 015716 042703 177700      BIC      #177700,R3
3211 015722 142777 000077 175374      BICB     #77,@ML.REG+150
3212 015730 150377 175370      BISB     R3,@ML.REG+150
3213 015734 116103 013434      MOVB     REM.TBL(R1),R3
3214 015740 042703 177700      BIC      #177700,R3
3215 015744 142777 000077 175362      BICB     #77,@ML.REG+160
3216 015752 150377 175356      BISB     R3,@ML.REG+160
3217 015756 000207      RTS      PC
3218
3219      ; Routine Size: 55 words
3220      ; Maximum stack depth per invocation: 4 words
3225
3226

```

2567

2569

2570

2569

2572

2573

2572

2575

2576

2577

2578

2539

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (12)

3228 :ML4
3229 :
3230 :
3231 :
3232 :
3233 :
3234 :
3235 :
3236 :
3237 :
3238 :
3239 :
3240 :
3241 :
3242 :
3243 :
3244 :
3245 :
3246 :
3247 :
3248 :
3249 :
3250 :
3251 :
3252 :
3253 :
3254 :
3255 :
3256 :
3257 :
3258 :
3259 :
3260 :
3261 :
3262 :
3263 :
3264 :
3265 :
3266 :
3267 :
3268 :
3269 :
3273 :

```

2580 routine FOR_ECC_ERR (SGL, UNC) : novalue =
2581   begin
2582
2583   !++
2584   FUNCTIONAL DESCRIPTION:
2585   'FORCE ECC ERROR' WHEN CALLED WILL FORCE
2586   VIA ECC DIAG MODE, SINGLE BIT ECC ERRORS
2587   OR UNCORRECTABLE ECC ERRORS INTO THE
2588   ECC DIAGNOSTIC REGISTERS
2589
2590   FORMAL PARAMETERS:
2591   SGL
2592   WHEN SET TO A ONE THIS
2593   ROUTINE WILL FORCE SINGLE BIT ERRORS
2594
2595   UNC
2596   WHEN SET TO A ONE THIS ROUTINE WILL
2597   FORCE UNCORRECTABLE ECC ERRORS
2598
2599   IMPLICIT INPUTS:
2600   ---
2601
2602   ECC_DM = ONE;
2603
2604   if .SGL
2605   then
2606     begin
2607       PAR_CRC_WRD = %b'111111';
2608       CRC_A = %b'000000';
2609       CRC_B = %b'000000';
2610     end
2611   else
2612     begin
2613       PAR_CRC_WRD = %b'111111';
2614       CRC_A = %b'111111';
2615       CRC_B = %b'000000';
2616     end;
2617
2618   end;

```

```

!SET ECC DIAG MODE
!IF SGL IS TRUE
!THEN FORCE A SINGLE BIT ERROR
!ELSE FORCE A UNCORRECTABLE ERROR

```

3277	015760				FOR.ECC.ERR:				
3278	015760	152777	000001	175306	BISB	#1,@ML.REG+120	:		2602
3279	015766	032766	000001	000004	BIT	#1,4(SP)	:	*,SGL	2604
3280	015774	001407			BEQ	1\$:		
3281	015776	052777	037400	175320	BIS	#37400,@ML.REG+150	:		2607


```
3283          :ML4
3284          :
3285
3286 016004 142777 000077 175312      BICB #77,@ML.REG+150
3287 016012 000406                BR    2$
3288 016014 052777 037400 175302 1$:  BIS #37400,@ML.REG+150
3289 016022 152777 000077 175274      BISB #77,@ML.REG+150
3290 016030 142777 000077 175276 2$:  BICB #77,@ML.REG+160
3291 016036 000207                RTS   PC
3292
3293          : Routine Size: 24 words
3294          : Maximum stack depth per invocation: 0 words
3299
3300
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

2608
2609
2613
2614
2615
2580

22-Dec-1980 09:24:31
 22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>BL3ML4.BLI.2 (13)

3302 :ML4
 3303 :
 3304 :
 3305 :
 3306 :
 3307 :
 3308 :
 3309 :
 3310 :
 3311 :
 3312 :
 3313 :
 3314 :
 3315 :
 3316 :
 3317 :
 3318 :
 3319 :
 3320 :
 3321 :
 3322 :
 3323 :
 3324 :
 3325 :
 3329 :
 3333 016040
 3334 016040
 3335 016044
 3336 016052
 3337 016060
 3338 :
 3339 :
 3340 :
 3345 :
 3346 :

2619 routine FIRST_BLK_XFER : novalue =

2620
 2621
 2622
 2623
 2624
 2625
 2626
 2627
 2628
 2629
 2630
 2631
 2632
 2633
 2634
 2635
 2636
 2637
 2638
 2639

!++
 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

2636
 2637
 2638
 2619

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>3L3ML4.BLI.2 (14)

3348 :ML4
3349 :
3350 :
3351 :
3352 :
3353 :
3354 :
3355 :
3356 :
3357 :
3358 :
3359 :
3360 :
3361 :
3362 :
3363 :
3364 :
3365 :
3366 :
3367 :
3368 :
3369 :
3370 :
3371 :
3372 :
3373 :
3374 :
3378 :
3382 :
3383 :
3384 :
3385 :
3386 :
3387 :
3388 :
3389 :
3390 :
3395 :
3396 :

2640 routine GD_BLK_XFER : novalue =

2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663

```

++
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
    
```

2659
2660
2661
2662
2640

```

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (15)

3398 :ML4

3399 :

3400 :

3401 :

3402 :

3403 :

3404 :

3405 :

3406 :

3407 :

3408 :

3409 :

3410 :

3411 :

3412 :

3413 :

3414 :

3415 :

3416 :

3417 :

3418 :

3419 :

3420 :

3421 :

3422 :

3423 :

3427 :

3431

3432

3433

3434

3435

3436

3437

3438

3439

3444

3445

2664 routine LAST_BLK_XFER : novalue =

2665

2666

2667

2668

2669

2670

2671

2672

2673

2674

2675

2676

2677

2678

2679

2680

2681

2682

2683

2684

2685

2686

```

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

```

LOADS A MASS BUS BLOCK
TRANSFER AT THE LAST BLOCK
ADDRESS
    
```

```

begin
ECC DIS = ONE;
MLDA = .LST.BLK;
MLBA = IO.BUF;
MLWC = not 255;
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 COMPLEMENT 255
    
```

LAST.BLK.XFER:

```

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

2682
2683
2684
2685
2664

```

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (16)

```

3447 :ML4
3448 :
3449 :
3450 : 2687 routine TST_LNG_WRD (NIB_NUM, NIB_PAT, ERR_FLG) : novalue =
3451 : 2688     begin
3452 : 2689
3453 : 2690 !++
3454 : 2691 ! FUNCTIONAL DESCRIPTION:
3455 : 2692 ! COMPARES THE CURRENT NIBBLE
3456 : 2693 ! POSITION IN 'NIB_SAVE' WITH THE
3457 : 2694 ! CURRENT TEST PATTERN. IF THE
3458 : 2695 ! TWO VALUES ARE NOT EQUAL AN
3459 : 2696 ! ERROR FLG IS SET WHICH THE
3460 : 2697 ! CALLER CAN INTERIGATE
3461 : 2698
3462 : 2699 ! FORMAL PARAMETERS:
3463 : 2700 ! NIB_NUM
3464 : 2701 ! CASE SELECT EXPRESSION TO SELECT THE
3465 : 2702 ! CURRENT NIBBLE TO BE EXAMINED
3466 : 2703
3467 : 2704 ! NIB_PAT
3468 : 2705 ! CURRENT NIBBLE PATTERN TO BE
3469 : 2706 ! COMPARED
3470 : 2707
3471 : 2708 ! ERR_FLG
3472 : 2709 ! CONTAINS THE ADDRESS (PASSED BY REF)
3473 : 2710 ! OF THE CALLERS ERROR FLG
3474 : 2711 ! TO ENABLE THE CALLER TO EXAMINE
3475 : 2712 ! THE ERROR STATUS OF THE ROUTINE CALL
3476 : 2713
3477 : 2714 ! IMPLICIT INPUTS:
3478 : 2715 ! NIB_SAVE
3479 : 2716 ! BLOCK OF 3 WORDS TO STORE
3480 : 2717 ! THE DATA FOUND IN MLD1, MLD2
3481 : 2718 ! AND MLE2 AFTER A DIAGNOSTIC MODE
3482 : 2719 ! READ
3483 : 2720
3484 : 2721 ! IMPLICIT OUTPUTS:  NONE
3485 : 2722
3486 : 2723
3487 : 2724     .ERR_FLG = ZERO;                                !CLEAR THE ERROR FLAG BACK IN THE CALLING ROUTINE
3488 : 2725
3489 : 2726 case .(NIB_NUM) from 0 to 9 of                        !SELECT THE NIBBLE TO BE TESTED
3490 : 2727     set
3491 : 2728     [0] :
3492 : 2729         if .NIB_SAVE [NIB_0] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3493 : 2730
3494 : 2731
3495 : 2732
3496 : 2733
3497 : 2734
3498 : 2735     [1] :
3499 : 2736         if .NIB_SAVE [NIB_1] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3500 : 2737
3501 : 2738

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (16)

```
3503 :ML4
3504 :
3505 :
3506 :      2739      !TEST NIBBLE 1 AND SET ERR FLG IF NEQ
3507 :      2740
3508 :      2741      [2] :
3509 :      2742      if .NIB_SAVE [NIB_2] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3510 :      2743
3511 :      2744      !TEST NIBBLE 2 AND SET ERR FLG IF NEQ
3512 :      2745
3513 :      2746
3514 :      2747      [3] :
3515 :      2748      if .NIB_SAVE [NIB_3] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3516 :      2749
3517 :      2750      !TEST NIBBLE 3 AND SET ERR FLG IF NEQ
3518 :      2751
3519 :      2752
3520 :      2753      [4] :
3521 :      2754      if .NIB_SAVE [NIB_4] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3522 :      2755
3523 :      2756      !TEST NIBBLE 4 AND SET ERR FLG IF NEQ
3524 :      2757
3525 :      2758
3526 :      2759      [5] :
3527 :      2760      if .NIB_SAVE [NIB_5] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3528 :      2761
3529 :      2762      !TEST NIBBLE 5 AND SET ERR FLG IF NEQ
3530 :      2763
3531 :      2764
3532 :      2765      [6] :
3533 :      2766      if .NIB_SAVE [NIB_6] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3534 :      2767
3535 :      2768      !TEST NIBBLE 6 AND SET ERR FLG IF NEQ
3536 :      2769
3537 :      2770
3538 :      2771      [7] :
3539 :      2772      if .NIB_SAVE [NIB_7] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3540 :      2773
3541 :      2774      !TEST NIBBLE 7 AND SET ERR FLG IF NEQ
3542 :      2775
3543 :      2776
3544 :      2777      [8] :
3545 :      2778      if .NIB_SAVE [NIB_8] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;
3546 :      2779
3547 :      2780      !TEST NIBBLE 8 AND SET ERR FLG IF NEQ
3548 :      2781
3549 :      2782
3550 :      2783      [9] :
3551 :      2784      if .NIB_SAVE [NIB_9] neq .(NIB_PAT)<0, 3>
3552 :      2785      then
3553 :      2786      .ERR_FLG = ONE
3554 :      2787      !TEST NIBBLE 9 AND SET ERR FLG IF NEQ
3555 :      2788      tes;
3556 :      2789
3557 :      2790      end;
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (16)

3559	;ML4								
3560	;								
3561									
3565									
3569	016146			TST.LNG.WRD:					
3570	016146	004167	165672	JSR	R1,\$SAVE2			2687	
3571	016152	016600	000010	MOV	10(SP),R0	:	ERR.FLG,*	2724	
3572	016156	005010		CLR	(R0)				
3573	016160	016601	000014	MOV	14(SP),R1	:	NIB.NUM,*	2726	
3574	016164	006301		ASL	R1				
3575	016166	066107	016172	ADD	1\$(R1),PC				
3576	016172	000024		1\$:	.WORD	2\$-1\$			
3577	016174	000052			.WORD	3\$-1\$			
3578	016176	000110			.WORD	4\$-1\$			
3579	016200	000140			.WORD	5\$-1\$			
3580	016202	000200			.WORD	6\$-1\$			
3581	016204	000226			.WORD	7\$-1\$			
3582	016206	000264			.WORD	8\$-1\$			
3583	016210	000314			.WORD	9\$-1\$			
3584	016212	000354			.WORD	10\$-1\$			
3585	016214	000404			.WORD	11\$-1\$			
3586	016216	016602	000012	2\$:	MOV	12(SP),R2	:	NIB.PAT,*	2731
3587	016222	042702	177760		BIC	#177760,R2			
3588	016226	016701	172720		MOV	NIB.SAVE,R1			
3589	016232	042701	177760		BIC	#177760,R1			
3590	016236	020102			CMP	R1,R2			
3591	016240	001577			BEQ	13\$			
3592	016242	000574			BR	12\$			
3593	016244	016601	000012	3\$:	MOV	12(SP),R1	:	NIB.PAT,*	2737
3594	016250	042701	177760		BIC	#177760,R1			
3595	016254	016702	172672		MOV	NIB.SAVE,R2			
3596	016260	006202			ASR	R2			
3597	016262	006202			ASR	R2			
3598	016264	006202			ASR	R2			
3599	016266	006202			ASR	R2			
3600	016270	042702	177760		BIC	#177760,R2			
3601	016274	020201			CMP	R2,R1			
3602	016276	001560			BEQ	13\$			
3603	016300	000555			BR	12\$			
3604	016302	016601	000012	4\$:	MOV	12(SP),R1	:	NIB.PAT,*	2743
3605	016306	042701	177760		BIC	#177760,R1			
3606	016312	016702	172634		MOV	NIB.SAVE,R2			
3607	016316	000302			SWAB	R2			
3608	016320	042702	177760		BIC	#177760,R2			
3609	016324	020201			CMP	R2,R1			
3610	016326	001544			BEQ	13\$			
3611	016330	000541			BR	12\$			
3612	016332	016601	000012	5\$:	MOV	12(SP),R1	:	NIB.PAT,*	2749

3614				:ML4				
3615				:				
3616								
3617	016336	042701	177760		BIC	#177760,R1		
3618	016342	016702	172604		MOV	NIB.SAVE,R2		
3619	016346	006202			ASR	R2		
3620	016350	006202			ASR	R2		
3621	016352	006202			ASR	R2		
3622	016354	006202			ASR	R2		
3623	016356	000302			SWAB	R2		
3624	016360	042702	177760		BIC	#177760,R2		
3625	016364	020201			CMP	R2,R1		
3626	016366	001524			BEQ	13\$		
3627	016370	000521			BR	12\$		
3628	016372	016601	000012	6\$:	MOV	12(SP),R1	; NIB.PAT,*	2755
3629	016376	042701	177760		BIC	#177760,R1		
3630	016402	016702	172546		MOV	NIB.SAVE+2,R2		
3631	016406	042702	177760		BIC	#177760,R2		
3632	016412	020201			CMP	R2,R1		
3633	016414	001511			BEQ	13\$		
3634	016416	000506			BR	12\$		
3635	016420	016601	000012	7\$:	MOV	12(SP),R1	; NIB.PAT,*	2761
3636	016424	042701	177760		BIC	#177760,R1		
3637	016430	016702	172520		MOV	NIB.SAVE+2,R2		
3638	016434	006202			ASR	R2		
3639	016436	006202			ASR	R2		
3640	016440	006202			ASR	R2		
3641	016442	006202			ASR	R2		
3642	016444	042702	177760		BIC	#177760,R2		
3643	016450	020201			CMP	R2,R1		
3644	016452	001472			BEQ	13\$		
3645	016454	000467			BR	12\$		
3646	016456	016601	000012	8\$:	MOV	12(SP),R1	; NIB.PAT,*	2767
3647	016462	042701	177760		BIC	#177760,R1		
3648	016466	016702	172462		MOV	NIB.SAVE+2,R2		
3649	016472	000302			SWAB	R2		
3650	016474	042702	177760		BIC	#177760,R2		
3651	016500	020201			CMP	R2,R1		
3652	016502	001456			BEQ	13\$		
3653	016504	000453			BR	12\$		
3654	016506	016601	000012	9\$:	MOV	12(SP),R1	; NIB.PAT,*	2773
3655	016512	042701	177760		BIC	#177760,R1		
3656	016516	016702	172432		MOV	NIB.SAVE+2,R2		
3657	016522	006202			ASR	R2		
3658	016524	006202			ASR	R2		
3659	016526	006202			ASR	R2		
3660	016530	006202			ASR	R2		
3661	016532	000302			SWAB	R2		
3662	016534	042702	177760		BIC	#177760,R2		
3663	016540	020201			CMP	R2,R1		
3664	016542	001436			BEQ	13\$		
3665	016544	000433			BR	12\$		
3666	016546	016601	000012	10\$:	MOV	12(SP),R1	; NIB.PAT,*	2779
3667	016552	042701	177760		BIC	#177760,R1		
3668	016556	016702	172374		MOV	NIB.SAVE+4,R2		

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

```
3670      ;ML4
3671      ;
3672
3673 016562 000302      SWAB      R2
3674 016564 042702 177760 BIC      #177760,R2
3675 016570 020201      CMP      R2,R1
3676 016572 001422      BEQ      13$
3677 016574 000417      BR       12$
3678 016576 016601 000012 11$: MOV   12(SP),R1      ; NIB.PAT,*      2785
3679 016602 042701 177770 BIC      #177770,R1
3680 016606 016702 172344 MOV      NIB.SAVE+4,R2
3681 016612 006202      ASR      R2
3682 016614 006202      ASR      R2
3683 016616 006202      ASR      R2
3684 016620 006202      ASR      R2
3685 016622 000302      SWAB      R2
3686 016624 042702 177770 BIC      #177770,R2
3687 016630 020201      CMP      R2,R1
3688 016632 001402      BEQ      13$
3689 016634 012710 000001 12$: MOV   #1,(R0)      ;
3690 016640 000207 13$: RTS   PC          ;
3691
3692      ; Routine Size: 158 words
3693      ; Maximum stack depth per invocation: 3 words
3698
3699
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (17)

```
3701 :ML4
3702 :
3703 :
3704 : 2791 routine XOR_LNG_WRD (NIB_NUM, NIB_PAT, RESULT) : novalue =
3705 : 2792     begin
3706 : 2793
3707 : 2794     !++
3708 : 2795     FUNCTIONAL DESCRIPTION:
3709 : 2796     EXCLUSIVE ORS THE CURRENT
3710 : 2797     NIBBLE POSITION IN 'NIB_SAVE'
3711 : 2798     WITH THE CURRENT TEST PATTERN
3712 : 2799     AND ASSIGNS THE RESULTS TO THE
3713 : 2800     FORMAL PARAMETER 'RESULT'.
3714 : 2801
3715 : 2802     FORMAL PARAMETERS:
3716 : 2803     NIB_NUM
3717 : 2804     CASE SELECT EXPRESSION TO
3718 : 2805     SELECT THE CURRENT NIBBLE TO BE
3719 : 2806     EXAMINED
3720 : 2807
3721 : 2808     NIB_PAT
3722 : 2809     CURRENT NIBBLE PATTERN TO BE
3723 : 2810     XOR'ED
3724 : 2811
3725 : 2812     RESULT
3726 : 2813     CONTAINS THE ADDRESS (PASSED BY REF)
3727 : 2814     OF AN OWN STORAGE LOCATION TO
3728 : 2815     ENABLE THE CALLER TO EXAMINE THE XOR RESULTS.
3729 : 2816
3730 : 2817     IMPLICIT INPUTS:
3731 : 2818     NIB_SAVE
3732 : 2819     BLOCK OF 3 WORDS TO STORE
3733 : 2820     THE DATA FOUND IN MLD1
3734 : 2821     MLD2 AND MLE2 AFTER A
3735 : 2822     DIAGNOSTIC MODE READ.
3736 : 2823
3737 : 2824     IMPLICIT OUTPUTS:  NONE
3738 : 2825
3739 : 2826
3740 : 2827     case .(NIB_NUM) from 0 to 9 of
3741 : 2828     set
3742 : 2829
3743 : 2830     [0] :
3744 : 2831     .RESULT = .NIB_SAVE [NIB_0] xor .NIB_PAT;
3745 : 2832     !XOR NIBBLE 0 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3746 : 2833
3747 : 2834     [1] :
3748 : 2835     .RESULT = .NIB_SAVE [NIB_1] xor .NIB_PAT;
3749 : 2836     !XOR NIBBLE 1 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3750 : 2837
3751 : 2838     [2] :
3752 : 2839     .RESULT = .NIB_SAVE [NIB_2] xor .NIB_PAT;
3753 : 2840     !XOR NIBBLE 2 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3754 : 2841
3755 : 2842     [3] :
```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (17)

```

3757 :ML4
3758 :
3759 :
3760 : 2843 .RESULT = .NIB_SAVE [NIB_3] xor .NIB PAT;
3761 : 2844 !XOR NIBBLE 3 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3762 : 2845
3763 : 2846 [4] :
3764 : 2847 .RESULT = .NIB_SAVE [NIB_4] xor .NIB PAT;
3765 : 2848 !XOR NIBBLE 4 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3766 : 2849
3767 : 2850 [5] :
3768 : 2851 .RESULT = .NIB_SAVE [NIB_5] xor .NIB PAT;
3769 : 2852 !XOR NIBBLE 5 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3770 : 2853
3771 : 2854 [6] :
3772 : 2855 .RESULT = .NIB_SAVE [NIB_6] xor .NIB PAT;
3773 : 2856 !XOR NIBBLE 6 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3774 : 2857
3775 : 2858 [7] :
3776 : 2859 .RESULT = .NIB_SAVE [NIB_7] xor .NIB PAT;
3777 : 2860 !XOR NIBBLE 7 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3778 : 2861
3779 : 2862 [8] :
3780 : 2863 .RESULT = .NIB_SAVE [NIB_8] xor .NIB PAT;
3781 : 2864 !XOR NIBBLE 8 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3782 : 2865
3783 : 2866 [9] :
3784 : 2867 .RESULT = .NIB_SAVE [NIB_9] xor .NIB PAT;
3785 : 2868 !XOR NIBBLE 9 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
3786 : 2869 tes;
3787 : 2870
3788 : 2871 end;
3792 :

```

```

3796 016642 XOR.LNG.WRD:
3797 016642 004167 165230 JSR R1,$SAVE4 ; 2791
3798 016646 016601 000016 MOV 16(SP),R1 ; NIB.PAT,* 2831
3799 016652 016602 000014 MOV 14(SP),R2 ; RESULT,*
3800 016656 016600 000020 MOV 20(SP),R0 ; NIB.NUM,* 2827
3801 016662 006300 ASL R0
3802 016664 066007 016670 ADD 1$(R0),PC
3803 016670 000024 1$: .WORD 2$-1$
3804 016672 000036 .WORD 3$-1$
3805 016674 000062 .WORD 4$-1$
3806 016676 000070 .WORD 5$-1$
3807 016700 000076 .WORD 6$-1$
3808 016702 000120 .WORD 8$-1$
3809 016704 000136 .WORD 9$-1$
3810 016706 000144 .WORD 10$-1$

```

```

3812      :ML4
3813      :
3814
3815 016710 000162      .WORD 12$-1$
3816 016712 000176      .WORD 15$-1$
3817 016714 016712 172232 2$: MOV NIB.SAVE,(R2) ; 2831
3818 016720 042712 177760 BIC #177760,(R2)
3819 016724 000472 BR 17$
3820 016726 016703 172220 3$: MOV NIB.SAVE,R3 ; 2835
3821 016732 006203 ASR R3
3822 016734 006203 ASR R3
3823 016736 006203 ASR R3
3824 016740 006203 ASR R3
3825 016742 042703 177760 BIC #177760,R3
3826 016746 010312 MOV R3,(R2)
3827 016750 000412 BR 7$
3828 016752 016704 172174 4$: MOV NIB.SAVE,R4 ; 2839
3829 016756 000437 BR 13$
3830 016760 016704 172166 5$: MOV NIB.SAVE,R4 ; 2843
3831 016764 000425 BR 11$
3832 016766 016712 172162 6$: MOV NIB.SAVE+2,(R2) ; 2847
3833 016772 042712 177760 BIC #177760,(R2)
3834 016776 010104 7$: MOV R1,R4
3835 017000 041204 BIC (R2),R4
3836 017002 040112 BIC R1,(R2)
3837 017004 050412 BIS R4,(R2)
3838 017006 000207 RTS PC ; 2827
3839 017010 016704 172140 8$: MOV NIB.SAVE+2,R4 ; 2851
3840 017014 006204 ASR R4
3841 017016 006204 ASR R4
3842 017020 006204 ASR R4
3843 017022 006204 ASR R4
3844 017024 000415 BR 14$
3845 017026 016704 172122 9$: MOV NIB.SAVE+2,R4 ; 2855
3846 017032 000411 BR 13$
3847 017034 016704 172114 10$: MOV NIB.SAVE+2,R4 ; 2859
3848 017040 006204 11$: ASR R4
3849 017042 006204 ASR R4
3850 017044 006204 ASR R4
3851 017046 006204 ASR R4
3852 017050 000402 BR 13$
3853 017052 016704 172100 12$: MOV NIB.SAVE+4,R4 ; 2863
3854 017056 000304 13$: SWAB R4
3855 017060 042704 177760 14$: BIC #177760,R4
3856 017064 000411 BR 16$
3857 017066 016704 172064 15$: MOV NIB.SAVE+4,R4 ; 2867
3858 017072 006204 ASR R4
3859 017074 006204 ASR R4
3860 017076 006204 ASR R4
3861 017100 006204 ASR R4
3862 017102 000304 SWAB R4
3863 017104 042704 177770 BIC #177770,R4
3864 017110 010412 16$: MOV R4,(R2)
3865 017112 010103 17$: MOV R1,R3
3866 017114 041203 BIC (R2),R3

```

3868
3869
3870
3871 017116 040112
3872 017120 050312
3873 017122 000207
3874
3875
3876
3881
3882

:ML4
:

BIC R1,(R2)
BIS R3,(R2)
RTS PC

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

2791

: Routine Size: 89 words
: Maximum stack depth per invocation: 5 words

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (18)

3884 :ML4
3885 :
3886 :
3887 :
3888 :
3889 :
3890 :
3891 :
3892 :
3893 :
3894 :
3895 :
3896 :
3897 :
3898 :
3899 :
3900 :
3901 :
3902 :
3903 :
3904 :
3905 :
3906 :
3907 :
3908 :
3909 :
3910 :
3911 :
3912 :
3913 :
3914 :
3915 :
3916 :
3917 :
3918 :
3919 :
3920 :
3921 :
3922 :
3923 :
3924 :
3925 :
3926 :
3927 :
3928 :
3929 :
3930 :
3931 :
3932 :
3933 :
3934 :
3935 :
3936 :
3937 :
3938 :

```

2872 routine LD_LNG_WRD (NIB_NUM, NIB_PAT) : novalue =
2873   begin
2874
2875   !++
2876   FUNCTIONAL DESCRIPTION:
2877     LOADS 'NIB_SAVE' WITH UNIQUE
2878     NIBBLE PATTERNS PRIOR TO WRITING
2879     TO MLD1, MLD2 AND MLE2
2880     DATA DIAGNOSTIC REGISTERS.
2881
2882   FORMAL PARAMETERS:
2883     NIB_NUM
2884     CASE SELECT EXPRESSION TO SELECT
2885     THE CURRENT NIBBLE TO BE LOADED
2886
2887     NIB_PAT
2888     CURRENT NIBBLE PATTERN TO BE
2889     LOADED
2890
2891   IMPLICIT INPUTS:
2892     NIB_SAVE
2893     BLOCK OF 3 WORDS TO STORE
2894     THE DATA TO BE WRITTEN
2895     INTO MLD1 MLD2 MLE2
2896
2897   IMPLICIT OUTPUTS:
2898     NIB_SAVE IS LOADED WITH
2899     THE CURRENT NIBBLE PATTERN
2900
2901
2902   case .(NIB_NUM) from 0 to 9 of
2903     set
2904
2905     [0] :
2906         NIB_SAVE [NIB_0] = .NIB_PAT;
2907
2908     [1] :
2909         NIB_SAVE [NIB_1] = .NIB_PAT;
2910
2911     [2] :
2912         NIB_SAVE [NIB_2] = .NIB_PAT;
2913
2914     [3] :
2915         NIB_SAVE [NIB_3] = .NIB_PAT;
2916
2917     [4] :
2918         NIB_SAVE [NIB_4] = .NIB_PAT;
2919
2920     [5] :
2921         NIB_SAVE [NIB_5] = .NIB_PAT;
2922
2923     [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-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
 22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (18)

```

3940 :ML4
3941 :
3942 :
3943 :      2924      NIB_SAVE [NIB_6] = .NIB_PAT;      !LOAD NIBBLE 6 WITH NIB_PAT
3944 :      2925
3945 :      2926      [7] :      NIB_SAVE [NIB_7] = .NIB_PAT;      !LOAD NIBBLE 7 WITH NIB_PAT
3946 :      2927
3947 :      2928
3948 :      2929      [8] :      NIB_SAVE [NIB_8] = .NIB_PAT;      !LOAD NIBBLE 8 WITH NIB_PAT
3949 :      2930
3950 :      2931
3951 :      2932      [9] :      NIB_SAVE [NIB_9] = .NIB_PAT      !LOAD NIBBLE 9 WITH NIB_PAT
3952 :      2933
3953 :      2934      tes;
3954 :      2935
3955 :      2936      end;
  
```

```

3963 017124      LD.LNG.WRD:
3964 017124 010146      MOV      R1,-(SP)      ;
3965 017126 016600 000004      MOV      4(SP),R0      ; NIB.PAT,*
3966 017132 016601 000006      MOV      6(SP),R1      ; NIB.NUM,*
3967 017136 006301      ASL      R1
3968 017140 066107 017144      ADD      1$(R1),PC
3969 017144 000024      1$:      .WORD      2$-1$
3970 017146 000040      .WORD      3$-1$
3971 017150 000064      .WORD      4$-1$
3972 017152 000102      .WORD      5$-1$
3973 017154 000134      .WORD      6$-1$
3974 017156 000150      .WORD      7$-1$
3975 017160 000174      .WORD      8$-1$
3976 017162 000212      .WORD      9$-1$
3977 017164 000244      .WORD     10$-1$
3978 017166 000262      .WORD     11$-1$
3979 017170 042700 177760      2$:      BIC      #177760,R0      ;
3980 017174 142767 000017 171750      BICB     #17,NIB.SAVE
3981 017202 000433      BR      6$
3982 017204 006300      3$:      ASL      R0      ;
3983 017206 006300      ASL      R0
3984 017210 006300      ASL      R0
3985 017212 006300      ASL      R0
3986 017214 042700 177417      BIC      #177417,R0
3987 017220 142767 000360 171724      BICB     #360,NIB.SAVE
3988 017226 000421      BR      6$
3989 017230 000300      4$:      SWAB     R0      ;
3990 017232 042700 170377      BIC      #170377,R0
3991 017236 042767 007400 171706      BICB     #7400,NIB.SAVE
3992 017244 000412      BR      6$
3993 017246 000300      5$:      SWAB     R0      ;
  
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

3995      :ML4
3996      :
3997
3998 017250 006300      ASL      R0
3999 017252 006300      ASL      R0
4000 017254 006300      ASL      R0
4001 017256 006300      ASL      R0
4002 017260 042700 007777  BIC      #7777,R0
4003 017264 042767 170000 171660  BIC      #170000,NIB.SAVE
4004 017272 050067 171654 6$:      BIS      R0,NIB.SAVE
4005 017276 000467      BR      15$      :
4006 017300 042700 177760 7$:      BIC      #177760,R0      :
4007 017304 142767 000017 171642  BICB     #17,NIB.SAVE+2      :
4008 017312 000433      BR      11$      :
4009 017314 006300 8$:      ASL      R0      :
4010 017316 006300      ASL      R0
4011 017320 006300      ASL      R0
4012 017322 006300      ASL      R0
4013 017324 042700 177417  BIC      #177417,R0
4014 017330 142767 000360 171616  BICB     #360,NIB.SAVE+2
4015 017336 000421      BR      11$
4016 017340 000300 9$:      SWAB     R0      :
4017 017342 042700 170377  BIC      #170377,R0      :
4018 017346 042767 007400 171600  BIC      #7400,NIB.SAVE+2
4019 017354 000412      BR      11$
4020 017356 000300 10$:     SWAB     R0      :
4021 017360 006300      ASL      R0
4022 017362 006300      ASL      R0
4023 017364 006300      ASL      R0
4024 017366 006300      ASL      R0
4025 017370 042700 007777  BIC      #7777,R0
4026 017374 042767 170000 171552  BIC      #170000,NIB.SAVE+2
4027 017402 050067 171546 11$:     BIS      R0,NIB.SAVE+2
4028 017406 000423      BR      15$      :
4029 017410 000300 12$:     SWAB     R0      :
4030 017412 042700 170377  BIC      #170377,R0
4031 017416 042767 007400 171532  BIC      #7400,NIB.SAVE+4
4032 017424 000412      BR      14$
4033 017426 000300 13$:     SWAB     R0      :
4034 017430 006300      ASL      R0
4035 017432 006300      ASL      R0
4036 017434 006300      ASL      R0
4037 017436 006300      ASL      R0
4038 017440 042700 107777  BIC      #107777,R0
4039 017444 042767 070000 171504  BIC      #70000,NIB.SAVE+4
4040 017452 050067 171500 14$:     BIS      R0,NIB.SAVE+4
4041 017456 012601 15$:     MOV      (SP)+,R1      :
4042 017460 000207      RTS      PC
4043
4044
4045

```

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

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (19)

4051 :ML4
4052 :
4053 :
4054 :
4055 :
4056 :
4057 :
4058 :
4059 :
4060 :
4061 :
4062 :
4063 :
4064 :
4065 :
4066 :
4067 :
4068 :
4069 :
4070 :
4071 :
4072 :
4073 :
4074 :
4078 :

2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957

```
routine WRT_CS1 (TST_PAT, index) : novalue =
begin
++
FUNCTIONAL DESCRIPTION:
    LOADS THE CONTROL & STATUS REGISTER 1 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.
```

MLCS1 = WRT_MASK;
end;

!LOAD GENERATED WRT_MASK PATTERN INTO MLCS1

4082	017462	004167	164356	WRT_CS1:JSR	R1,\$SAVE2	:	2937
4083	017466	016600	000010	MOV	10(SP),R0	: INDEX,*	2956
4084	017472	006300		ASL	R0		
4085	017474	006300		ASL	R0		
4086	017476	006300		ASL	R0		
4087	017500	010001		MOV	R0,R1		
4088	017502	016100	013156	MOV	ML,REG+2(R1),R0		
4089	017506	056600	000012	BIS	12(SP),R0	: TST.PAT,*	
4090	017512	046100	013160	BIC	ML,REG+4(R1),R0		
4091	017516	016102	013162	MOV	ML,REG+6(R1),R2		
4092	017522	050002		BIS	R0,R2		
4093	017524	010277	173424	MOV	R2,@ML.REG		
4094	017530	000207		RTS	PC	:	2937
4095							
4096							
4097							
4102							
4103							

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>B.3ML4.BLI.2 (20)

4105 :ML4
4106 :
4107 :
4108 :
4109 :
4110 :
4111 :
4112 :
4113 :
4114 :
4115 :
4116 :
4117 :
4118 :
4119 :
4120 :
4121 :
4122 :
4123 :
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 :

```
2958 routine RD_CS1 (TST_PAT, index, ERR_FLG) : novalue =
2959   begin
2960
2961   !++
2962   FUNCTIONAL DESCRIPTION:
2963
2964   COMPARES THE CONTENTS OF THE
2965   CONTROL & STATUS REGISTER 1
2966   WITH THE MASKED DATA PATTERN
2967   GENERATED BY THE MACRO 'WRT_MASK'
2968
2969   IF THE COMPARE IS NOT EQUAL THEN
2970   THE FORMAL PARAMETER 'ERR_FLG' IS
2971   ASSIGNED A ONE TO INDICATE THE
2972   ERROR.
2973
2974   FORMAL PARAMETERS:
2975
2976   TST PAT
2977   DATA PATTERN TO BE MASKED AND
2978   COMPARED AGAINST THE CONTENTS
2979   OF THE REGISTER UNDER TEST.
2980
2981   INDEX
2982   USED BY THE MACRO WRT_MASK TO
2983   SELECT THE CURRENT REGISTER'S ADDRESS,
2984   FORCED HI, FORCED LO AND DON'T CARE
2985   MASK INFORMATION.
2986
2987   ERR_FLG
2988   CONTAINS THE ADDRESS (PASSED BY REF)
2989   OF THE CALLER'S ERROR_FLG TO ENABLE THE
2990   CALLER TO EXAMINE THE ERROR STATUS
2991   OF THE ROUTINE CALL.
2992
2993   IMPLICIT INPUTS:
2994   WT_DATA
2995   GETS LOADED WITH THE GENERATED
2996   WRT_MASK DATA PATTERN THUS ALLOWING
2997   CALLER TO PRINT FAILING GOOD DATA.
2998
2999   RD_DATA
3000   GETS LOADED WITH DATA READ FROM THE
3001   REGISTER THUS ALLOWING CALLER
3002   TO PRINT FAILING BAD DATA.
3003
3004   IMPLICIT OUTPUTS:
3005   GLOBAL LOCATION WR_DATA
3006   AND RD_DATA LOADED WITH GOOD
3007   AND BAD REGISTER DATA
3008
3009
```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3M .BLI.2 (20)

4161 :ML4
4162 :
4163 :
4164 : 3010
4165 : 3011
4166 : 3012
4167 : 3013
4168 : 3014
4169 : 3015
4170 : 3016
4171 : 3017

```
.ERR_FLG = ZERO;
WT_DATA = WRT_MASK;
RD_DATA = .MLCS1 or .IGNORE;

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

end;

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

!READ THE REG FOR WRT_MASK

!SET ERROR FLAG IF NEQ
```

4179	017532	004167	164306	RD.CS1:	JSR	R1,\$SAVE2	:	2958
4180	017536	005076	000010		CLR	@10(SP)	: ERR.FLG	3010
4181	017542	016600	000012		MOV	12(SP),R0	: INDEX,*	3011
4182	017546	006300			ASL	R0		
4183	017550	006300			ASL	R0		
4184	017552	006300			ASL	R0		
4185	017554	010001			MOV	R0,R1		
4186	017556	016100	013156		MOV	ML.REG+2(R1),R0		
4187	017562	056600	000014		BIS	14(SP),R0	: TST.PAT,*	
4188	017566	046100	013160		BIC	ML.REG+4(R1),R0		
4189	017572	016102	013162		MOV	ML.REG+6(R1),R2		
4190	017576	050002			BIS	R0,R2		
4191	017600	010267	173322		MOV	R2,WRT_DATA		
4192	017604	017702	173344		MOV	@ML.REG,R2	:	3012
4193	017610	056102	013162		BIS	ML.REG+6(R1),R2		
4194	017614	010267	173310		MOV	R2,RD_DATA		
4195	017620	026767	173302	173302	CMP	WT_DATA,RD_DATA	:	3014
4196	017626	001403			BEQ	1\$		
4197	017630	012776	000001	000010	MOV	#1,@10(SP)	: *,ERR.FLG	
4198	017636	000207			1\$: RTS	PC	:	2958

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

4199
4200
4201
4206
4207

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (21)

```

4209 :ML4
4210 :
4211 :
4212 : 3018 routine WRT_ER (TST_PAT, index) : novalue =
4213 : 3019     begin
4214 : 3020
4215 : 3021     !++
4216 : 3022     FUNCTIONAL DESCRIPTION:
4217 : 3023
4218 : 3024         LOADS THE ERROR REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
4219 : 3025
4220 : 3026     FORMAL PARAMETERS:
4221 : 3027
4222 : 3028         TST_PAT
4223 : 3029         CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
4224 : 3030
4225 : 3031         INDEX
4226 : 3032         USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTER'S ADDRESS.
4227 : 3033         FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
4228 : 3034     !--
4229 : 3035
4230 : 3036     MLER = WRT_MASK;
4231 : 3037     end;
4235 :

```

!LOAD GENERATE WRT_MASK PATTERN INTO MLER

```

4239 017640 004167 164200      WRT.ER: JSR      R1,$SAVE2           ;
4240 017644 016600 000010      MOV      10(SP),R0       ; INDEX,*
4241 017650 006300              ASL      R0
4242 017652 006300              ASL      R0
4243 017654 006300              ASL      R0
4244 017656 010001              MOV      R0,R1
4245 017660 016100 013156      MOV      ML.REG+2(R1),R0
4246 017664 056600 000012      BIS      12(SP),R0       ; TST.PAT,*
4247 017670 046100 013160      BIC      ML.REG+4(R1),R0
4248 017674 016102 013162      MOV      ML.REG+6(R1),R2
4249 017700 050002              BIS      R0,R2
4250 017702 010277 173326      MOV      R2,@ML.REG+60
4251 017706 000207              RTS      PC
4252 :
4253 :
4254 :
4259 :
4260 :

```

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (22)

```
4262 :ML4
4263 :
4264 :
4265 : 3038 routine RD_ER (TST_PAT, index, ERR_FLG) : novalue =
4266 : 3039     begin
4267 : 3040
4268 : 3041     +-
4269 : 3042     FUNCTIONAL DESCRIPTION:
4270 : 3043
4271 : 3044     COMPARES THE CONTENTS OF THE
4272 : 3045     ERROR REGISTER WITH
4273 : 3046     THE MASKED DATA PATTERN
4274 : 3047     GENERATED BY THE MACRO 'WRT_MASK'.
4275 : 3048
4276 : 3049     IF THE COMPARE IS NOT EQUAL THEN
4277 : 3050     THE FORMAL PARAMETER 'ERR_FLG' IS
4278 : 3051     ASSIGNED A ONE TO INDICATE THE
4279 : 3052     ERROR.
4280 : 3053
4281 : 3054     FORMAL PARAMETERS:
4282 : 3055
4283 : 3056     TST PAT
4284 : 3057     DATA PATTERN TO BE MASKED AND
4285 : 3058     COMPARED AGAINST THE CONTENTS
4286 : 3059     OF THE REGISTER UNDER TEST.
4287 : 3060
4288 : 3061     INDEX
4289 : 3062     USED BY THE MACRO WRT_MASK TO
4290 : 3063     SELECT THE CURRENT REGISTER ADDRESS,
4291 : 3064     FORCED HI, FORCED LO AND DON'T CARE
4292 : 3065     MASK INFORMATION.
4293 : 3066
4294 : 3067     ERR_FLG
4295 : 3068     CONTAINS THE ADDRESS (PASSED BY REF)
4296 : 3069     OF THE CALLERS ERROR_FLG TO ENABLE THE
4297 : 3070     CALLER TO EXAMINE THE ERROR STATUS
4298 : 3071     OF THE ROUTINE CALL.
4299 : 3072
4300 : 3073     IMPLICIT INPUTS:
4301 : 3074     WT_DATA
4302 : 3075     GETS LOADED WITH THE GENERATED
4303 : 3076     WRT_MASK DATA PATTERN THUS ALLOWING
4304 : 3077     CALLER TO PRINT FAILING GOOD DATA.
4305 : 3078
4306 : 3079     RD_DATA
4307 : 3080     GETS LOADED WITH DATA READ FROM THE
4308 : 3081     REGISTER THUS ALLOWING CALLER
4309 : 3082     TO PRINT FAILING BAD DATA.
4310 : 3083
4311 : 3084     IMPLICIT OUTPUTS:
4312 : 3085     GLOBAL LOCATION WR_DATA
4313 : 3086     AND RD_DATA LOADED WITH GOOD
4314 : 3087     AND BAD REGISTER DATA
4315 : 3088
4316 : 3089     --
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (22)

```

4318 :ML4
4319 :
4320 :
4321 : 3090
4322 : 3091 .ERR_FLG = ZERO; !CLEAR THE ERROR FLAG
4323 : 3092 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
4324 : 3093 RD_DATA = .MLER or .IGNORE; !READ AND SAVE THE REGISTER
4325 : 3094
4326 : 3095 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
4327 : 3096
4328 : 3097 !SET ERROR FLAG IF NEQ
4329 : 3098
4330 :
4331 :
4332 :
4333 :
4337 017710 004167 164130 RD.ER: JSR R1,$SAVE2 ; 3038
4338 017714 005076 000010 CLR @10(SP) ; ERR.FLG 3091
4339 017720 016600 000012 MOV 12(SP),R0 ; INDEX,* 3092
4340 017724 006300 ASL R0
4341 017726 006300 ASL R0
4342 017730 006300 ASL R0
4343 017732 010001 MOV R0,R1
4344 017734 016100 013156 MOV ML.REG+2(R1),R0
4345 017740 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
4346 017744 046100 013160 BIC ML.REG+4(R1),R0
4347 017750 016102 013162 MOV ML.REG+6(R1),R2
4348 017754 050002 BIS R0,R2
4349 017756 010267 173144 MOV R2,WRT_DATA
4350 017762 017702 173246 MOV @ML.REG+60,R2 ; 3093
4351 017766 056102 013162 BIS ML.REG+6(R1),R2
4352 017772 010267 173132 MOV R2,RD_DATA
4353 017776 026767 173124 173124 CMP WRT_DATA,RD_DATA ; 3095
4354 020004 001403 BEQ 1$
4355 020006 012776 000001 000010 MOV #1,@10(SP) ; *.ERR.FLG
4356 020014 000207 1$: RTS PC ; 3038
4357
4358 ; Routine Size: 35 words
4359 ; Maximum stack depth per invocation: 3 words
4364
4365

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (23)

```

4367 :ML4
4368 :
4369 :
4370 : 3099 routine WRT_DA (TST_PAT, index) : novalue =
4371 : 3100 begin
4372 : 3101
4373 : 3102 !++
4374 : 3103 FUNCTIONAL DESCRIPTION:
4375 : 3104
4376 : 3105 LOADS THE DESIRED SECTOR WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
4377 : 3106
4378 : 3107 FORMAL PARAMETERS:
4379 : 3108
4380 : 3109 TST_PAT
4381 : 3110 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
4382 : 3111
4383 : 3112 INDEX
4384 : 3113 USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
4385 : 3114 FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
4386 : 3115
4387 : 3116
4388 : 3117 MLDA = WRT_MASK;
4389 : 3118 end;

```

!LOAD MLDA WITH GENERATED WRT_MASK PATTERN

```

4397 020016 004167 164022 WRT.DA: JSR R1,$SAVE2 ;
4398 020022 016600 000010 MOV 10(SP),R0 ; INDEX,*
4399 020026 006300 ASL R0
4400 020030 006300 ASL R0
4401 020032 006300 ASL R0
4402 020034 010001 MOV R0,R1
4403 020036 016100 013156 MOV ML.REG+2(R1),R0
4404 020042 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
4405 020046 046100 013160 BIC ML.REG+4(R1),R0
4406 020052 016102 013162 MOV ML.REG+6(R1),R2
4407 020056 050002 BIS R0,R2
4408 020060 010277 173120 MOV R2,@ML.REG+30
4409 020064 000207 RTS PC ;
4410
4411 ; Routine Size: 20 words
4412 ; Maximum stack depth per invocation: 3 words
4417
4418

```

3099
3117

3099

4420 :ML4
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 :
4453 :
4454 :
4455 :
4456 :
4457 :
4458 :
4459 :
4460 :
4461 :
4462 :
4463 :
4464 :
4465 :
4466 :
4467 :
4468 :
4469 :
4470 :
4471 :
4472 :
4473 :
4474 :

```
3119 routine RD_DA (TST_PAT, index, ERR_FLG) : novalue =  
3120 begin  
3121  
3122 !++  
3123 FUNCTIONAL DESCRIPTION:  
3124  
3125 COMPARES THE CONTENTS OF THE  
3126 DESIRED SECTOR ADDRESS REGISTER  
3127 WITH THE MASKED DATA PATTERN  
3128 GENERATED BY THE MACRO 'WRT_MASK'  
3129  
3130 IF THE COMPARE IS NOT EQUAL THEN  
3131 THE FORMAL PARAMETER 'ERR_FLG' IS  
3132 ASSIGNED A ONE TO INDICATE THE  
3133 ERROR  
3134  
3135 FORMAL PARAMETERS:  
3136  
3137 TST_PAT  
3138 DATA PATTERN TO BE MASKED AND  
3139 COMPARED AGAINST THE CONTENTS  
3140 OF THE REGISTER UNDER TEST  
3141  
3142 INDEX  
3143 USED BY THE MACRO WRT_MASK TO  
3144 SELECT THE CURRENT REGISTER'S ADDRESS,  
3145 FORCED HI, FORCED LO AND DON'T CARE  
3146 MASK INFORMATION  
3147  
3148 ERR_FLG  
3149 CONTAINS THE ADDRESS (PASSED BY REF)  
3150 OF THE CALLER'S ERROR_FLG TO ENABLE THE  
3151 CALLER TO EXAMINE THE ERROR STATUS  
3152 OF THE ROUTINE CALL.  
3153 IMPLICIT INPUTS:  
3154 WT_DATA  
3155 GETS LOADED WITH THE GENERATED  
3156 WRT_MASK DATA PATTERN THUS ALLOWING  
3157 CALLER TO PRINT FAILING GOOD DATA.  
3158  
3159 RD_DATA  
3160 GETS LOADED WITH DATA READ FROM THE  
3161 REGISTER THUS ALLOWING CALLER  
3162 TO PRINT FAILING BAD DATA.  
3163  
3164 IMPLICIT OUTPUTS:  
3165 GLOBAL LOCATION WR_DATA  
3166 AND RD_DATA LOADED WITH GOOD  
3167 AND BAD REGISTER DATA  
3168  
3169  
3170
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (24)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (24)

```

4476 ;ML4
4477 :
4478 :
4479 : 3171 .ERR_FLG = ZERO; !CLEAR THE ERROR FLAG
4480 : 3172 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
4481 : 3173 RD_DATA = .MLDA or .IGNORE; !READ AND SAVE THE REGISTER
4482 : 3174
4483 : 3175 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
4484 : 3176
4485 : 3177 !SET ERROR FLAG IF NEQ
4486 : 3178 end;
4490
4494 020066 004167 163752 RD.DA: JSR R1,$SAVE2 ; 3119
4495 020072 005076 000010 CLR @10(SP) ; ERR.FLG 3171
4496 020076 016600 000012 MOV 12(SP),R0 ; INDEX,* 3172
4497 020102 006300 ASL R0
4498 020104 006300 ASL R0
4499 020106 006300 ASL R0
4500 020110 010001 MOV R0,R1
4501 020112 016100 013156 MOV ML.REG+2(R1),R0
4502 020116 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
4503 020122 046100 013160 BIC ML.REG+4(R1),R0
4504 020126 016102 013162 MOV ML.REG+6(R1),R2
4505 020132 050002 BIS R0,R2
4506 020134 010267 172766 MOV R2,WT_DATA
4507 020140 017702 173040 MOV @ML.REG+30,R2 ; 3173
4508 020144 056102 013162 BIS ML.REG+6(R1),R2
4509 020150 010267 172754 MOV R2,RD_DATA
4510 020154 026767 172746 172746 CMP WT_DATA,RD_DATA ; 3175
4511 020162 001403 BEQ 1$
4512 020164 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
4513 020172 000207 1$: RTS PC ; 3119
4514
4515 ; Routine Size: 35 words
4516 ; Maximum stack depth per invocation: 3 words
4521
4522

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:KNEALE>BL3ML4.BLI.2 (25)

```

4524 :ML4
4525 :
4526 :
4527 : 3179 routine WRT_MR (TST_PAT, index) : novalue =
4528 : 3180 begin
4529 : 3181
4530 : 3182 !++
4531 : 3183 FUNCTIONAL DESCRIPTION:
4532 : 3184
4533 : 3185 LOADS THE MAINTENANCE REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
4534 : 3186
4535 : 3187 FORMAL PARAMETERS:
4536 : 3188
4537 : 3189 TST_PAT
4538 : 3190 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
4539 : 3191
4540 : 3192 INDEX
4541 : 3193 USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTER'S ADDRESS,
4542 : 3194 FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
4543 : 3195 !--
4544 : 3196
4545 : 3197 MLMR = WRT_MASK; !LOAD MLMR WITH GENERATED WRT_MASK PATTERN
4546 : 3198 end;

```

```

4554 020174 004167 163644 WRT.MR: JSR R1,$SAVE2 ; 3179
4555 020200 016600 000010 MOV 10(SP),R0 ; INDEX,* 3197
4556 020204 006300 ASL R0
4557 020206 006300 ASL R0
4558 020210 006300 ASL R0
4559 020212 010001 MOV R0,R1
4560 020214 016100 013156 MOV ML.REG+2(R1),R0
4561 020220 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
4562 020224 046100 013160 BIC ML.REG+4(R1),R0
4563 020230 016102 013162 MOV ML.REG+6(R1),R2
4564 020234 050002 BIS R0,R2
4565 020236 010277 173032 MOV R2,@ML.REG+120
4566 020242 000207 RTS PC ; 3179
4567 :
4568 : ; Routine Size: 20 words
4569 : ; Maximum stack depth per invocation: 3 words
4574 :
4575 :

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (26)

```
4577 :ML4
4578 :
4579 :
4580 : 3199 routine RD_MR (TST_PAT, index, ERR_FLG) : novalue =
4581 : 3200 begin
4582 : 3201
4583 : 3202 !++
4584 : 3203 FUNCTIONAL DESCRIPTION:
4585 : 3204
4586 : 3205 COMPARES THE CONTENTS OF THE
4587 : 3206 MAINTENANCE REGISTER WITH THE
4588 : 3207 MASKED DATA PATTERN
4589 : 3208 GENERATED BY THE MACRO 'WRT_MASK'.
4590 : 3209
4591 : 3210 IF THE COMPARE IS NOT EQUAL THEN
4592 : 3211 THE FORMAL PARAMETER 'ERR_FLG' IS
4593 : 3212 ASSIGNED A ONE TO INDICATE THE
4594 : 3213 ERROR.
4595 : 3214
4596 : 3215 FORMAL PARAMETERS:
4597 : 3216
4598 : 3217 TST PAT
4599 : 3218 DATA PATTERN TO BE MASKED AND
4600 : 3219 COMPARED AGAINST THE CONTENTS
4601 : 3220 OF THE REGISTER UNDER TEST.
4602 : 3221
4603 : 3222 INDEX
4604 : 3223 USED BY THE MACRO WRT_MASK TO
4605 : 3224 SELECT THE CURRENT REGISTERS ADDRESS,
4606 : 3225 FORCED HI, FORCED LO AND DON'T CARE
4607 : 3226 MASK INFORMATION.
4608 : 3227
4609 : 3228 ERR_FLG
4610 : 3229 CONTAINS THE ADDRESS (PASSED BY REF)
4611 : 3230 OF THE CALLERS ERROR_FLG TO ENABLE THE
4612 : 3231 CALLER TO EXAMINE THE ERROR STATUS
4613 : 3232 OF THE ROUTINE CALL.
4614 : 3233
4615 : 3234 IMPLICIT INPUTS:
4616 : 3235 WT_DATA
4617 : 3236 GETS LOADED WITH THE GENERATED
4618 : 3237 WRT_MASK DATA PATTERN THUS ALLOWING
4619 : 3238 CALLER TO PRINT FAILING GOOD DATA.
4620 : 3239
4621 : 3240 RD_DATA
4622 : 3241 GETS LOADED WITH DATA READ FROM THE
4623 : 3242 REGISTER THUS ALLOWING CALLER
4624 : 3243 TO PRINT FAILING BAD DATA.
4625 : 3244
4626 : 3245 IMPLICIT OUTPUTS:
4627 : 3246 GLOBAL LOCATION WR_DATA
4628 : 3247 AND RD_DATA LOADED WITH GOOD
4629 : 3248 AND BAD REGISTER DATA
4630 : 3249
4631 : 3250 !--
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (26)

```

4633 ;ML4
4634 :
4635 :
4636 : 3251
4637 : 3252
4638 : 3253
4639 : 3254
4640 : 3255
4641 : 3256
4642 : 3257
4643 : 3258
4644 : 3259
4648 :
4652 020244 004167 163574 RD.MR: JSR R1,$SAVE2 ;
4653 020250 005076 000010 CLR @10(SP) ; ERR.FLG
4654 020254 016600 000012 MOV 12(SP),R0 ; INDEX,*
4655 020260 006300 ASL R0
4656 020262 006300 ASL R0
4657 020264 006300 ASL R0
4658 020266 010001 MOV R0,R1
4659 020270 016100 013156 MOV ML.REG+2(R1),R0
4660 020274 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
4661 020300 046100 013160 BIC ML.REG+4(R1),R0
4662 020304 016102 013162 MOV ML.REG+6(R1),R2
4663 020310 050002 BIS R0,R2
4664 020312 010267 172610 MOV R2,WT.DATA
4665 020316 017702 172752 MOV @ML.REG+120,R2 ;
4666 020322 056102 013162 BIS ML.REG+6(R1),R2 ; 3254
4667 020326 010267 172576 MOV R2,RD.DATA
4668 020332 026767 172570 172570 CMP WT.DATA,RD.DATA ;
4669 020340 001403 BEQ 1$ ; 3256
4670 020342 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
4671 020350 000207 1$: RTS PC ; 3199
4672 :
4673 : ; Routine Size: 35 words
4674 : ; Maximum stack depth per invocation: 3 words
4679 :
4680 :

```

```

.ERR_FLG = ZERO; !CLEAR THE ERROR FLAG
WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
RD_DATA = .MLR 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

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML 4.BLI.2 (27)

```
4682 :ML4
4683 :
4684 :
4685 : 3260 routine WRT_PA (TST_PAT, index) : novalue =
4686 : 3261 begin
4687 : 3262
4688 : 3263 !++
4689 : 3264 FUNCTIONAL DESCRIPTION:
4690 : 3265 LOADS THE PROM ADDRESS REGISTER WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
4691 : 3266 FORMAL PARAMETERS:
4692 : 3267 TST_PAT
4693 : 3268 CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
4694 : 3269 INDEX
4695 : 3270 USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
4696 : 3271 FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
4697 : 3272
4698 : 3273
4699 : 3274 PROM_DIS = ONE;
4700 : 3275 MLPA = WRT_MASK;
4701 : 3276 PROM_DIS = ZERO;
4702 : 3277 end;
```

```
!SET PROM DISABLE BIT
!LOAD MLPA WITH GENERATED WRT_MASK PATTERN
!CLEAR PROM DISABLE BIT
```

```
4710 020352 004167 163466 WRT.PA: JSR R1,$SAVE2 ; 3260
4711 020356 152777 000040 172710 BISB #40,@ML.REG+i20 ; 3274
4712 020364 016600 000010 MOV 10(SP),R0 ; INDEX,* 3275
4713 020370 006300 ASL R0
4714 020372 006300 ASL R0
4715 020374 006300 ASL R0
4716 020376 010001 MOV R0,R1
4717 020400 016100 013156 MOV ML.REG+2(R1),R0
4718 020404 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
4719 020410 046100 013160 BIC ML.REG+4(R1),R0
4720 020414 016102 013162 MOV ML.REG+6(R1),R2
4721 020420 050002 BIS R0,R2
4722 020422 010277 172626 MOV R2,@ML.REG+100
4723 020426 142777 000040 172640 BICB #40,@ML.REG+120 ; 3276
4724 020434 000207 RTS PC ; 3260
```

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

4735 :ML4
4736 :
4737 :
4738 :
4739 :
4740 :
4741 :
4742 :
4743 :
4744 :
4745 :
4746 :
4747 :
4748 :
4749 :
4750 :
4751 :
4752 :
4753 :
4754 :
4755 :
4756 :
4757 :
4758 :
4759 :
4760 :
4761 :
4762 :
4763 :
4764 :
4765 :
4766 :
4767 :
4768 :
4769 :
4770 :
4771 :
4772 :
4773 :
4774 :
4775 :
4776 :
4777 :
4778 :
4779 :
4780 :
4781 :
4782 :
4783 :
4784 :
4785 :
4786 :
4787 :
4788 :
4789 :

3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329

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

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
FROM 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 ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

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.

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-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (28)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (28)

```

4791 :ML4
4792 :
4793 :
4794 :      3330      .ERR_FLG = ZERO;          !CLEAR THE ERROR FLAG
4795 :      3331      PROM_DIS = ONE;          !SET THE PROM DISABLE BIT
4796 :      3332      WT_DATA = WRT_MASK;      !SAVE THE DATA WRITTEN TO THE REGISTER
4797 :      3333      RD_DATA = .MLPA or .IGNORE; !READ AND SAVE THE REGISTER
4798 :      3334
4799 :      3335      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;      !READ THE REG FOR WRT_MASK
4800 :      3336
4801 :      3337      PROM_DIS = ZERO;          !SET ERROR IF NEQ
4802 :      3338      end;                    !CLEAR THE PROM DISABLE BIT
4806
4810 020436 004167 163402      RD.PA: JSR      R1,$SAVE2          ;
4811 020442 005076 000010      CLR      @10(SP)          ; ERR.FLG
4812 020446 152777 000040 172620  BISB     #40,@ML.REG+120 ;
4813 020454 016600 000012      MOV      12(SP),R0      ; INDEX,*
4814 020460 006300      ASL      R0
4815 020462 006300      ASL      R0
4816 020464 006300      ASL      R0
4817 020466 010001      MOV      R0,R1
4818 020470 016100 013156      MOV      ML.REG+2(R1),R0
4819 020474 056600 000014      BIS      14(SP),R0      ; TST.PAT,*
4820 020500 046100 013160      BIC      ML.REG+4(R1),R0
4821 020504 016102 013162      MOV      ML.REG+6(R1),R2
4822 020510 050002      BIS      R0,R2
4823 020512 010267 172410      MOV      R2,WT_DATA
4824 020516 017702 172532      MOV      @ML.REG+100,R2 ;
4825 020522 056102 013162      BIS      ML.REG+6(R1),R2 ;
4826 020526 010267 172376      MOV      R2,RD_DATA
4827 020532 026767 172370 172370  CMP      WT_DATA,RD_DATA ;
4828 020540 001403      BEQ
4829 020542 012776 000001 000010  MOV      #1,@10(SP)      ; *,ERR.FLG
4830 020550 142777 000040 172516 1$: BICB     #40,@ML.REG+120 ;
4831 020556 000207      RTS      PC          ;
4832
4833      ; Routine Size: 41 words
4834      ; Maximum stack depth per invocation: 3 words
4839
4840

```

3278
3330
3331
3332
3333
3335
3337
3278

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (29)

```

4842 :ML4
4843 :
4844 :
4845 : 3339 routine WRT_E1 (TST_PAT, index) : novalue =
4846 : 3340 begin
4847 : 3341
4848 : 3342
4849 : 3343
4850 : 3344
4851 : 3345
4852 : 3346
4853 : 3347
4854 : 3348
4855 : 3349
4856 : 3350
4857 : 3351
4858 : 3352
4859 : 3353
4860 : 3354
4861 : 3355
4862 : 3356
4866 :
4870 020560 004167 163260 WRT.E1: JSR R1,$SAVE2 : 3339
4871 020564 152777 000001 172502 BISB #1,@ML.REG+120 : 3353
4872 020572 016600 000010 MOV 10(SP),R0 : INDEX,* 3354
4873 020576 006300 ASL R0
4874 020600 006300 ASL R0
4875 020602 006300 ASL R0
4876 020604 010001 MOV R0,R1
4877 020606 016100 013156 MOV ML.REG+2(R1),R0
4878 020612 056600 000012 BIS 12(SP),R0 : TST.PAT,*
4879 020616 046100 013160 BIC ML.REG+4(R1),R0
4880 020622 016102 013162 MOV ML.REG+6(R1),R2
4881 020626 050002 BIS R0,R2
4882 020630 010277 172470 MOV R2,@ML.REG+150
4883 020634 142777 000001 172432 BICB #1,@ML.REG+120 : 3355
4884 020642 000207 RTS PC : 3339
4885 :
4886 : Routine Size: 26 words
4887 : Maximum stack depth per invocation: 3 words
4892 :
4893 :

```

```

!++
FUNCTIONAL DESCRIPTION:
LOADS THE ECC CRC WORD REG 1 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.

```

```

ECC_DM = ONE;           !SET ECC DIAG MODE
MLET = WRT_MASK;       !LOAD MLE1 WITH GENERATED WRT_MASK PATTERN
ECC_DM = ZERO;        !CLEAR ECC DIAG MODE
end;

```


4895 :ML4
4896 :
4897 :
4898 :
4899 :
4900 :
4901 :
4902 :
4903 :
4904 :
4905 :
4906 :
4907 :
4908 :
4909 :
4910 :
4911 :
4912 :
4913 :
4914 :
4915 :
4916 :
4917 :
4918 :
4919 :
4920 :
4921 :
4922 :
4923 :
4924 :
4925 :
4926 :
4927 :
4928 :
4929 :
4930 :
4931 :
4932 :
4933 :
4934 :
4935 :
4936 :
4937 :
4938 :
4939 :
4940 :
4941 :
4942 :
4943 :
4944 :
4945 :
4946 :
4947 :
4948 :
4949 :

3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
3399
3400
3401
3402
3403
3404
3405
3406
3407
3408

routine RD_E1 (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
ECC CRC WORD REGISTER 1
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 ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

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.

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

.ERR_FLG = ZERO;

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (30)

!CLEAP THE ERROR FLAG

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:(NEALE)BL3ML4.BLI.2 (30)

```

4951 :ML4
4952 :
4953 :
4954 :      3409      ECC_DM = ONE;           !SET ECC DIAG MODE
4955 :      3410      WT_DATA = WRT_MASK;       !SAVE THE DATA WRITTEN TO THE REGISTER
4956 :      3411      RD_DATA = .MLE1 or .IGNORE; !READ AND SAVE THE REGISTER
4957 :      3412
4958 :      3413      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;   !READ THE REG FOR WRT_MASK
4959 :      3414
4960 :      3415      ECC_DM = ZERO;           !SET ERROR FLAG IF NEQ
4961 :      3416      !CLEAR ECC DIAG MODE
4962 :      3417      end;
4966 :
4970 020644 004167 163174      RD.E1: JSR      R1,$SAVE2           ;
4971 020650 005076 000010      CLR      @10(SP)           ; ERR.FLG
4972 020654 152777 000001 172412  BISB     #1,@ML.REG+120     ;
4973 020662 016600 000012      MOV      12(SP),R0         ; INDEX,*
4974 020666 006300      ASL      R0
4975 020670 006300      ASL      R0
4976 020672 006300      ASL      R0
4977 020674 010001      MOV      R0,R1
4978 020676 016100 013156      MOV      ML.REG+2(R1),R0
4979 020702 056600 000014      BIS      14(SP),R0         ; TST.PAT,*
4980 020706 046100 013160      BIC      ML.REG+4(R1),R0
4981 020712 016102 013162      MOV      ML.REG+6(R1),R2
4982 020716 050002      BIS      R0,R2
4983 020720 010267 172202      MOV      R2,WT_DATA
4984 020724 017702 172374      MOV      @ML.REG+150,R2     ;
4985 020730 056102 013162      BIS      ML.REG+6(R1),R2
4986 020734 010267 172170      MOV      R2,RD_DATA
4987 020740 026767 172162 172162  CMP      WT_DATA,RD_DATA     ;
4988 020746 001403      BEQ
4989 020750 012776 000001 000010  MOV      #1,@10(SP)         ; *,ERR.FLG
4990 020756 142777 000001 172310 1$: BICB     #1,@ML.REG+120     ;
4991 020764 000207      RTS      PC                ;
4992 :
4993 :
4994 :
4999 :
5000 :

```

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (31)

5002 :ML4
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 :
5039 :

3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444
3445
3446
3447
3448

routine WRT_E2 (TST_PAT, index) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:

LOADS THE ECC CRC WORD
REGISTER 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.

ECC_DM = ONE;
MLE2_MASK = %o'177700';
MLE2 = WRT_MASK;
MLE2_MASK = %o'000000';
ECC_DM = ZERO;
end;

!SET ECC DIAG MODE
!MASK OUT DATA DIAG BITS
!LOAD MLE2 WITH GENERATED WRT_MASK PATTERN
!RESTORE MLE2_MASK
!CLEAR ECC DIAG MODE

5043	020766	004167	163052	WRT.E2:	JSR	R1,\$SAVE2	:	3418
5044	020772	152777	000001	172274	BISB	#1,@ML.REG+120	:	3443
5045	021000	012767	177700	172334	MOV	#-100,ML.REG+166	:	3444
5046	021006	016600	000010		MOV	10(SP),R0	:	3445
5047	021012	006300			ASL	R0	:	
5048	021014	006300			ASL	R0	:	
5049	021016	006300			ASL	R0	:	
5050	021020	010001			MOV	R0,R1	:	
5051	021022	016100	013156		MOV	ML.REG+2(R1),R0	:	
5052	021026	056600	000012		BIS	12(SP),R0	:	
5053	021032	046100	013160		BIC	ML.REG+4(R1),R0	:	
5054	021036	016102	013162		MOV	ML.REG+6(R1),R2	:	
5055	021042	050002			BIS	R0,R2	:	

5057
5058
5059
5060
5061
5062
5063
5064
5065
5066
5071
5072

;ML4
;

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

021044 010277 172264
021050 005067 172266
021054 142777 000001 172212
021062 000207

MOV R2,@ML.REG+160
CLR ML.REG+166
BICB #1,@ML.REG+120
RTS PC

;
;
;

3446
3447
3418

: Routine Size: 31 words
: Maximum stack depth per invocation: 3 words

5074 :ML4
 5075 :
 5076 :
 5077 :
 5078 :
 5079 :
 5080 :
 5081 :
 5082 :
 5083 :
 5084 :
 5085 :
 5086 :
 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 :

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
 3499
 3500

routine RD_E2 (TST_PAT, index, ERR_FLG) : novalue =
 begin

++
 FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
 ECC CRC WORD REGISTER 2
 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 ADDRESS,
 FORCED HI, FORCED LO AND DON'T CARE
 MASK INFORMATION.

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.

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-Dec-1980 09:24:31
 22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
 PA: <NEALE>BL3ML4.BLI.2 (32)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212) {
22-Dec-1980 09:21:22 PA:(NEALE)BL3ML4.BLI.2 (32)

```

5130 ;ML4
5131 :
5132 :
5133 : 3501
5134 : 3502 .ERR_FLG = ZERO; !CLEAR ERROR FLAG
5135 : 3503 ECC_DM = ONE; !SET ECC DIAG MODE
5136 : 3504 MLE2_MASK = %o'177700'; !MASK OUT DATA DIAG BITS
5137 : 3505 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5138 : 3506 RD_DATA = .MLE2 or .IGNORE; !READ AND SAVE THE REGISTER
5139 : 3507
5140 : 3508 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
5141 : 3509
5142 : 3510 !SET ERROR FLG IF NEQ
5143 : 3511 MLE2_MASK = %o'000000'; !RESTORE MLE2 MASK
5144 : 3512 ECC_DM = ZERO; !CLEAR ECC DIAG MODE
5145 : 3513 end;
5149 :
5153 021064 004167 162754 RD.E2: JSR R1,$SAVE2 ; 3449
5154 021070 005076 J00010 CLR @10(SP) ; ERR.FLG 3502
5155 021074 152777 000001 172172 BISB #1,@ML.REG+120 ; 3503
5156 021102 012767 177700 172232 MOV #-100,ML.REG+166 ; 3504
5157 021110 016600 000012 MOV 12(SP),R0 ; INDEX,* 3505
5158 021114 006300 ASL R0
5159 021116 006300 ASL R0
5160 021120 006300 ASL R0
5161 021122 010001 MOV R0,R1
5162 021124 016100 013156 MOV ML.REG+2(R1),R0
5163 021130 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5164 021134 046100 013160 BIC ML.REG+4(R1),R0
5165 021140 016102 013162 MOV ML.REG+6(R1),R2
5166 021144 050002 BIS R0,R2
5167 021146 010267 171754 MOV R2,WT_DATA
5168 021152 017702 172156 MOV @ML.REG+160,R2 ; 3506
5169 021156 056102 013162 BIS ML.REG+6(R1),R2
5170 021162 010267 171742 MOV R2,RD_DATA
5171 021166 026767 171734 171734 CMP WT_DATA,RD_DATA ; 3508
5172 021174 001403 BEQ 1$
5173 021176 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
5174 021204 005067 172132 1$: CLR ML.REG+166 ; 3511
5175 021210 142777 000001 172056 BICB #1,@ML.REG+120 ; 3512
5176 021216 000207 RTS PC ; 3449
5177 :
5178 :
5179 :
; Routine Size: 46 words
; Maximum stack depth per invocation: 3 words

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (33)

```

5185 :ML4
5186 :
5187 :
5188 : 3514 routine WRT_PD (TST_PAT, index) : novalue =
5189 : 3515     begin
5190 : 3516
5191 : 3517     ++
5192 : 3518     FUNCTIONAL DESCRIPTION:
5193 : 3519
5194 : 3520         LOADS THE PROM DATA
5195 : 3521         REGISTER WITH A DATA PATTERN
5196 : 3522         GENERATED BY THE MACRO
5197 : 3523         WRT_MASK
5198 : 3524
5199 : 3525     FORMAL PARAMETERS:
5200 : 3526
5201 : 3527         TST_PAT
5202 : 3528         CURRENT DATA PATTERN TO BE
5203 : 3529         LOADED IN THE REGISTER.
5204 : 3530
5205 : 3531         INDEX
5206 : 3532         USED BY THE MACRO WRT_MASK
5207 : 3533         TO SELECT THE CURRENT REGISTERS
5208 : 3534         ADDRESS, FORCED HI, FORCED
5209 : 3535         LO AND DON'T CARE MASK
5210 : 3536         INFORMATION
5211 : 3537
5212 : 3538     --
5213 : 3539
5214 : 3540     PROM_RW = ONE;
5215 : 3541     DAT_DM = ONE;
5216 : 3542     MLPD = WRT_MASK;
5217 : 3543     DAT_CLK = ONE;
5218 : 3544     PROM_RW = ZERO;
5219 : 3545     DAT_DM = ZERO;
5220 : 3546     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

```

```

5228 021220 004167 162620      WRT.PD: JSR      R1,$SAVE2          :
5229 021224 152777 000100 172042  BISB   #100,@ML.REG+120  :
5230 021232 152777 000010 172034  BISB   #10,@ML.REG+120  :
5231 021240 016600 000010          MOV    10(SP),R0        : INDEX,*
5232 021244 006300          ASL   R0
5233 021246 006300          ASL   R0
5234 021250 006300          ASL   R0
5235 021252 010001          MOV   R0,R1
5236 021254 016100 013156          MOV   ML.REG+2(R1),R0
5237 021260 056600 000012          BIS   12(SP),R0
5238 021264 046100 013160          BIC   ML.REG+4(R1),R0 ; TST.PAT,*

```

3514
3540
3541
3542

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

5240
5241
5242
5243 021270 016102 013162
5244 021274 050002
5245 021276 010277 172102
5246 021302 152777 000020 171764
5247 021310 142777 000100 171756
5248 021316 142777 000010 171750
5249 021324 000207
5250
5251
5252
5257
5258

:ML4
:

```
MOV ML.REG+6(R1),R2  
BIS R0,R2  
MOV R2,@ML.REG+230  
BISB #20,@ML.REG+120  
BICB #100,@ML.REG+120  
BICB #10,@ML.REG+120  
RTS PC
```

3543
3544
3545
3514

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

5260 :ML4
5261 :
5262 :
5263 :
5264 :
5265 :
5266 :
5267 :
5268 :
5269 :
5270 :
5271 :
5272 :
5273 :
5274 :
5275 :
5276 :
5277 :
5278 :
5279 :
5280 :
5281 :
5282 :
5283 :
5284 :
5285 :
5286 :
5287 :
5288 :
5289 :
5290 :
5291 :
5292 :
5293 :
5294 :
5295 :
5296 :
5297 :
5298 :
5299 :
5300 :
5301 :
5302 :
5303 :
5304 :
5305 :
5306 :
5307 :
5308 :
5309 :
5310 :
5311 :
5312 :
5313 :
5314 :

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (34)

```
3547 routine RD_PD (TST_PAT, index, ERR_FLG) : novalue =
3548   begin
3549
3550 !++
3551 ! FUNCTIONAL DESCRIPTION:
3552
3553     COMPARES THE CONTENTS OF THE
3554     PROM DATA REGISTER
3555     WITH THE MASKED DATA PATTERN
3556     GENERATED BY THE MACRO 'WRT_MASK'
3557
3558     IF THE COMPARE IS NOT EQUAL THEN
3559     THE FORMAL PARAMETER 'ERR_FLG' IS
3560     ASSIGNED A ONE TO INDICATE THE
3561     ERROR
3562
3563 ! FORMAL PARAMETERS:
3564
3565     TST_PAT
3566     DATA PATTERN TO BE MASKED AND
3567     COMPARED AGAINST THE CONTENTS
3568     OF THE REGISTER UNDER TEST
3569
3570     INDEX
3571     USED BY THE MACRO WRT_MASK TO
3572     SELECT THE CURRENT REGISTER ADDRESS.
3573     FORCED HI, FORCED LO AND DON'T CARE
3574     MASK INFORMATION.
3575
3576     ERR_FLG
3577     CONTAINS THE ADDRESS (PASSED BY REF)
3578     OF THE CALLERS ERROR_FLG TO ENABLE THE
3579     CALLER TO EXAMINE THE ERROR STATUS
3580     OF THE ROUTINE CALL.
3581
3582 ! IMPLICIT INPUTS:
3583     WT_DATA
3584     GETS LOADED WITH THE GENERATED
3585     WRT_MASK DATA PATTERN THUS ALLOWING
3586     CALLER TO PRINT FAILING GOOD DATA.
3587
3588     RD_DATA
3589     GETS LOADED WITH DATA READ FROM THE
3590     REGISTER THUS ALLOWING CALLER
3591     TO PRINT FAILING BAD DATA.
3592
3593 ! IMPLICIT OUTPUTS:
3594     GLOBAL LOCATION WR_DATA
3595     AND RD_DATA LOADED WITH GOOD
3596     AND BAD REGISTER DATA
3597
3598 !--
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (34)

```

5316 ;ML4
5317 :
5318 :
5319 : 3599
5320 : 3600 .ERR_FLG = ZERO; !CLEAR ERROR FLG
5321 : 3601 PROM_RW = ONE; !SET PROM READ WRITE
5322 : 3602 DAT_DM = ONE; !SET DATA DIAG MODE
5323 : 3603 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5324 : 3604 RD_DATA = .MLPD or .IGNORE; !READ AND SAVE THE REGISTER
5325 : 3605
5326 : 3606 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
5327 : 3607
5328 : 3608 !SET ERROR FLAG IF NEQ
5329 : 3609 PROM_RW = ZERO; !CLEAR PROM READ WRITE
5330 : 3610 DAT_DM = ZERO; !CLEAR DATA DIAG MODE
5331 : 3611 end;
5332 :
5333 :
5339 021326 004167 162512 RD.PD: JSR R1,$SAVE2 ;
5340 021332 005076 000010 CLR @10(SP) ; ERR.FLG
5341 021336 152777 000100 171730 BISB #100,@ML.REG+120 ;
5342 021344 152777 000010 171722 BISB #10,@ML.REG+120 ;
5343 021352 016600 000012 MOV 12(SP),R0 ; INDEX,*
5344 021356 006300 ASL R0 ;
5345 021360 006300 ASL R0 ;
5346 021362 006300 ASL R0 ;
5347 021364 010001 MOV R0,R1 ;
5348 021366 016100 013156 MOV ML.REG+2(R1),R0 ;
5349 021372 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5350 021376 046100 013160 BIC ML.REG+4(R1),R0 ;
5351 021402 016102 013162 MOV ML.REG+6(R1),R2 ;
5352 021406 050002 BIS R0,R2 ;
5353 021410 010267 171512 MOV R2,WT_DATA ;
5354 021414 017702 171764 MOV @ML.REG+230,R2 ; 3604
5355 021420 056102 013162 BIS ML.REG+6(R1),R2 ;
5356 021424 010267 171500 MOV R2,RD_DATA ;
5357 021430 026767 171472 171472 CMP WT_DATA,RD_DATA ; 3606
5358 021436 001403 BEQ 1$ ;
5359 021440 012776 000001 000010 MOV #1,@10(SP) ; *,ERR.FLG
5360 021446 142777 000100 171620 1$: BICB #100,@ML.REG+120 ; 3609
5361 021454 142777 000010 171612 BICB #10,@ML.REG+120 ; 3610
5362 021462 000207 RTS PC ; 3547
5363 :
5364 : ; Routine Size: 47 words
5365 : ; Maximum stack depth per invocation: 3 words

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (35)

```
5371 :ML4
5372 :
5373 :
5374 : 3612 routine WRT_EL (TST_PAT, index) : novalue =
5375 : 3613 begin
5376 : 3614
5377 : 3615 |**
5378 : 3616 | FUNCTIONAL DESCRIPTION:
5379 : 3617 | DUMMY ROUTINE CALL TO ASSIST IN THE READ
5380 : 3618 | WRITE REGISTER ALGORITHM
5381 : 3619 |
5382 : 3620 | FORMAL PARAMETERS:
5383 : 3621 | TST PAT
5384 : 3622 | DATA PATTERN TO BE MASKED AND
5385 : 3623 | COMPARED AGAINST THE CONTENTS
5386 : 3624 | OF THE REGISTER UNDER TEST
5387 : 3625 |
5388 : 3626 | INDEX
5389 : 3627 | USED BY THE MACRO WRT_MASK TO
5390 : 3628 | SELECT THE CURRENT REGISTER ADDRESS,
5391 : 3629 | FORCED HI, FORCED LO AND DON'T CARE
5392 : 3630 | MASK INFORMATION.
5393 : 3631 |
5394 : 3632 | ---
5395 : 3633 |
5396 : 3634 |
5397 : 3635 | ERROR LOCATION REG IS READ ONLY
5398 : 3636 | return;
5399 : 3637 | end;
5403 :
5407 021464 000207 WRT.EL: RTS PC ;
5408 :
5409 : ; Routine Size: 1 word
5410 : ; Maximum stack depth per invocation: 0 words
5415 :
5416 :
```

3612

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (36)

```
5418 :ML4
5419 :
5420 :
5421 : 3638 routine RD_EL (TST_PAT, index, ERR_FLG) : novalue =
5422 : 3639 begin
5423 : 3640
5424 : 3641 ++
5425 : 3642 FUNCTIONAL DESCRIPTION:
5426 : 3643
5427 : 3644 COMPARES THE CONTENTS OF THE
5428 : 3645 ERROR LOCATION REGISTER
5429 : 3646 WITH THE CONTENTS OF TST_PAT.
5430 : 3647
5431 : 3648 IF THE COMPARE IS NOT EQUAL THEN
5432 : 3649 THE FORMAL PARAMETER 'ERR FLG' IS
5433 : 3650 ASSIGNED A ONE TO INDICATE THE
5434 : 3651 ERROR
5435 : 3652
5436 : 3653 FORMAL PARAMETERS:
5437 : 3654 TST_PAT
5438 : 3655 DATA PATTERN TO BE MASKED AND
5439 : 3656 COMPARED AGAINST THE CONTENTS
5440 : 3657 OF THE REGISTER UNDER TEST
5441 : 3658
5442 : 3659 INDEX
5443 : 3660 USED BY THE MACRO WRT_MASK TO
5444 : 3661 SELECT THE CURRENT REGISTER'S ADDRESS,
5445 : 3662 FORCED HI, FORCED LO AND DON'T CARE
5446 : 3663 MASK INFORMATION.
5447 : 3664
5448 : 3665 ERR_FLG
5449 : 3666 CONTAINS THE ADDRESS (PASSED BY REF)
5450 : 3667 OF THE CALLER'S ERROR_FLG TO ENABLE THE
5451 : 3668 CALLER TO EXAMINE THE ERROR STATUS
5452 : 3669 OF THE ROUTINE CALL.
5453 : 3670
5454 : 3671 IMPLICIT INPUTS:
5455 : 3672 WT_DATA
5456 : 3673 GETS LOADED WITH THE GENERATED
5457 : 3674 WRT_MASK DATA PATTERN THUS ALLOWING
5458 : 3675 CALLER TO PRINT FAILING GOOD DATA.
5459 : 3676
5460 : 3677 RD_DATA
5461 : 3678 GETS LOADED WITH DATA READ FROM THE
5462 : 3679 REGISTER THUS ALLOWING CALLER
5463 : 3680 TO PRINT FAILING BAD DATA.
5464 : 3681
5465 : 3682 IMPLICIT OUTPUTS:
5466 : 3683 GLOBAL LOCATION WR_DATA
5467 : 3684 AND RD_DATA LOADED WITH GOOD
5468 : 3685 AND BAD REGISTER DATA
5469 : 3686
5470 : 3687
5471 : 3688
5472 : 3689 .ERR_FLG = ZERO;
```

5474 :ML4
5475 :
5476 :
5477 :
5478 :
5479 :
5480 :
5481 :
5482 :
5486 :
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5504
5505

3690
3691
3692
3693
3694
3695

WT_DATA = .TST.PAT;
RD_DATA = .MLEC;

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

end;

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (36)

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

!READ REGISTER FOR WT_DATA

RD.EL: CLR @2(SP) : ERR.FLG
MOV 6(SP),WT_DATA : TST.PAT,*
MOV @M.REG+220,RD_DATA :
CMP RD_DATA,WT_DATA :
BEQ 1\$:
MOV #1,@2(SP) : *,ERR.FLG
IS: RTS PC :

3689
3690
3691
3693
3638

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (37)

```
5507 :ML4  
5508 :  
5509 :  
5510 : 3696 routine WRT_EE (TST_PAT, index) : novalue =  
5511 : 3697 begin  
5512 : 3698  
5513 : 3699 !++  
5514 : 3700 ! FUNCTIONAL DESCRIPTION:  
5515 : 3701 ! DUMMY ROUTINE CALL TO ASSIST IN THE READ  
5516 : 3702 ! WRITE REGISTER ALGORITHM  
5517 : 3703  
5518 : 3704 ! FORMAL PARAMETERS:  
5519 : 3705 ! TST PAT  
5520 : 3706 ! DATA PATTERN TO BE MASKED AND  
5521 : 3707 ! COMPARED AGAINST THE CONTENTS  
5522 : 3708 ! OF THE REGISTER UNDER TEST  
5523 : 3709  
5524 : 3710 ! INDEX  
5525 : 3711 ! USED BY THE MACRO WRT_MASK TO  
5526 : 3712 ! SELECT THE CURRENT REGISTER ADDRESS,  
5527 : 3713 ! FORCED HI, FORCED LO AND DON'T CARE  
5528 : 3714 ! MASK INFORMATION.  
5529 : 3715  
5530 : 3716 !-  
5531 : 3717  
5532 : 3718 ! ECC ERROR REGISTER IS READ ONLY  
5533 : 3719 return;  
5534 : 3720 end;  
5538  
5542 021526 000207 WRT.EE: RTS PC ;  
5543  
5544 ; Routine Size: 1 word  
5545 ; Maximum stack depth per invocation: 0 words  
5550  
5551
```

3696

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (38)

5553 :ML4
5554 :
5555 :
5556 :
5557 :
5558 :
5559 :
5560 :
5561 :
5562 :
5563 :
5564 :
5565 :
5566 :
5567 :
5568 :
5569 :
5570 :
5571 :
5572 :
5573 :
5574 :
5575 :
5576 :
5577 :
5578 :
5579 :
5580 :
5581 :
5582 :
5583 :
5584 :
5585 :
5586 :
5587 :
5588 :
5589 :
5590 :
5591 :
5592 :
5593 :
5594 :
5595 :
5596 :
5597 :
5598 :
5599 :
5600 :
5601 :
5602 :
5603 :
5604 :
5605 :
5606 :
5607 :

```
3721 routine RD_EE (TST_PAT, index, ERR_FLG) : novalue =
3722 begin
3723
3724 !++
3725 FUNCTIONAL DESCRIPTION:
3726
3727 COMPARES THE CONTENTS OF THE
3728 ECC ERROR REGISTER WITH TST_PAT.
3729
3730 IF THE COMPARE IS NOT EQUAL THEN
3731 THE FORMAL PARAMETER 'ERR_FLG' IS
3732 ASSIGNED A ONE TO INDICATE THE
3733 ERROR.
3734
3735 FORMAL PARAMETERS:
3736 TST_PAT
3737 DATA PATTERN TO BE
3738 COMPARED AGAINST THE CONTENTS
3739 OF THE REGISTER UNDER TEST.
3740
3741 INDEX
3742 USED BY THE MACRO MLEE TO
3743 SELECT THE CURRENT REGISTER ADDRESS.
3744
3745 ERR_FLG
3746 CONTAINS THE ADDRESS (PASSED
3747 BY REF) OF THE CALLERS ERROR_FLG TO ENABLE
3748 THE CALLER TO EXAMINE THE ERROR STATUS
3749 OF THE ROUTINE CALL.
3750
3751 IMPLICIT INPUTS:
3752 WT_DATA
3753 GETS LOADED WITH THE TST_PAT
3754 THUS ALLOWING CALLER TO PRINT
3755 THE FAILING DATA.
3756
3757 RD_DATA
3758 GETS LOADED WITH DATA READ FROM THE
3759 REGISTER THUS ALLOWING CALLER
3760 TO PRINT FAILING BAD DATA.
3761
3762 IMPLICIT OUTPUTS:
3763 GLOBAL LOCATION WR_DATA
3764 AND RD_DATA LOADED WITH GOOD
3765 AND BAD REGISTER DATA
3766
3767
3768
3769 .ERR_FLG = ZERO;
3770 WT_DATA = .TST_PAT;
3771 RD_DATA = .MLEE;
3772
```

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

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (38)

```
5609 :ML4
5610 :
5611 :
5612 :      3773      if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;      !READ MLEE FOR TST_PAT
5613 :      3774
5614 :      3775      !SET THE ERROR FLAG IF NEQ
5615 :      3776      end;
5619 :
5623 021530 005076 000002      RD.EE: CLR      @2(SP)      : ERR.FLG      3769
5624 021534 016667 000006 171364      MOV      6(SP),WT.DATA      : TST.PAT,*      3770
5625 021542 017767 171616 171360      MOV      @ML.REG+210,RD.DATA      :      3771
5626 021550 026767 171354 171350      CMP      RD.DATA,WT.DATA      :      3773
5627 021556 001403      BEQ      1$
5628 021560 012776 000001 000002      MOV      #1,@2(SP)      : *.ERR.FLG
5629 021566 000207      1$: RTS      PC      :      3721
5630 :
5631 :      ; Routine Size: 16 words
5632 :      ; Maximum stack depth per invocation: 0 words
5637 :
5638 :
```


22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (39)

5640 :ML4
5641 :
5642 :
5643 :
5644 :
5645 :
5646 :
5647 :
5648 :
5649 :
5650 :
5651 :
5652 :
5653 :
5654 :
5655 :
5656 :
5657 :
5658 :
5659 :
5660 :
5664 :
5668 :
5669 :
5670 :
5671 :
5672 :
5673 :
5674 :
5675 :
5676 :
5677 :
5678 :
5679 :
5680 :
5681 :
5682 :
5683 :
5684 :
5685 :
5690 :
5691 :

```

3777 routine WRT_D1 (TST_PAT, index) : novalue =
3778     begin
3779
3780     !++
3781     FUNCTIONAL DESCRIPTION:
3782     LOADS THE DATA DIAG REG 1 WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
3783     FORMAL PARAMETERS:
3784     TST_PAT
3785     CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
3786     INDEX
3787     USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
3788     FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
3789     !--
3790
3791     DAT_DM = ONE;           !SET DATA DIAG MODE
3792     MLD1 = WRT_MASK;       !LOAD MLD1 WITH GENERATED WRT_MASK PATTERN
3793     DAT_DM = ZERO;        !CLEAR DATA DIAG MODE
3794     end;
    
```

```

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

```

3777
3791
3792
: INDEX,*
: TST.PAT,*
:
3793
3777
    
```

```

WRT.D1: JSR      R1,$SAVE2
        BISB    #10,@ML.REG+120
        MOV     10(SP),R0
        ASL    R0
        ASL    R0
        ASL    R0
        MOV    R0,R1
        MOV    ML.REG+2(R1),R0
        BIS   12(SP),R0
        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
        RTS   PC
    
```

5693 :ML4
5694 :
5695 :
5696 :
5697 :
5698 :
5699 :
5700 :
5701 :
5702 :
5703 :
5704 :
5705 :
5706 :
5707 :
5708 :
5709 :
5710 :
5711 :
5712 :
5713 :
5714 :
5715 :
5716 :
5717 :
5718 :
5719 :
5720 :
5721 :
5722 :
5723 :
5724 :
5725 :
5726 :
5727 :
5728 :
5729 :
5730 :
5731 :
5732 :
5733 :
5734 :
5735 :
5736 :
5737 :
5738 :
5739 :
5740 :
5741 :
5742 :
5743 :
5744 :
5745 :
5746 :
5747 :

```
3795 routine RD_D1 (TST_PAT, index, ERR_FLG) : novalue =  
3796 begin  
3797  
3798  
3799 ++  
3800 FUNCTIONAL DESCRIPTION:  
3801  
3802 COMPARES THE CONTENTS OF THE  
3803 DATA DIAG REGISTER 1  
3804 WITH THE MASKED DATA PATTERN  
3805 GENERATED BY THE MACRO 'WRT_MASK'.  
3806  
3807 IF THE COMPARE IS NOT EQUAL THEN  
3808 THE FORMAL PARAMETER 'ERR_FLG' IS  
3809 ASSIGNED A ONE TO INDICATE THE  
3810 ERROR.  
3811  
3812 FORMAL PARAMETERS:  
3813  
3814 TST_PAT  
3815 DATA PATTERN TO BE MASKED AND  
3816 COMPARED AGAINST THE CONTENTS  
3817 OF THE REGISTER UNDER TEST.  
3818  
3819 INDEX  
3820 USED BY THE MACRO WRT_MASK TO  
3821 SELECT THE CURRENT REGISTER ADDRESS,  
3822 FORCED HI, FORCED LO AND DON'T CARE  
3823 MASK INFORMATION.  
3824  
3825 ERR_FLG  
3826 CONTAINS THE ADDRESS (PASSED BY REF)  
3827 OF THE CALLERS ERROR FLG TO ENABLE THE  
3828 CALLER TO EXAMINE THE ERROR STATUS  
3829 OF THE ROUTINE CALL.  
3830  
3831 IMPLICIT INPUTS:  
3832 WT_DATA  
3833 GETS LOADED WITH THE GENERATED  
3834 WRT_MASK DATA PATTERN THUS ALLOWING  
3835 CALLER TO PRINT FAILING GOOD DATA.  
3836  
3837 RD_DATA  
3838 GETS LOADED WITH DATA READ FROM THE  
3839 REGISTER THUS ALLOWING CALLER  
3840 TO PRINT FAILING BAD DATA.  
3841  
3842 IMPLICIT OUTPUTS:  
3843 GLOBAL LOCATION WR_DATA  
3844 AND RD_DATA LOADED WITH GOOD  
3845 AND BAD REGISTER DATA  
3846
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (40)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (40)

```

5749 :ML4
5750 :
5751 :
5752 : 3847
5753 : 3848 .ERR_FLG = ZERO; !CLEAR THE ERROR FLG
5754 : 3849 DAT_DM = ONE; !SET DATA DIAG MODE
5755 : 3850 ML_FUNC = write; !LOAD MLCST WITH WRITE FUNCTION
5756 : 3851 DAT_CLK = ONE; !DO A DATA CLK
5757 : 3852
5758 : 3853 if .REG_INIT_FLG IS_SET !SEE IF CALLER IS REG INIT TEST
5759 : 3854 !SET ERROR FLAG IF NEQ
5760 : 3855 then
5761 : 3856 begin
5762 : 3857 CLR_MBUS; !CLEAR MBUS TO GENERATE INIT DATA
5763 : 3858 DAT_DM = ONE;
5764 : 3859 end;
5765 : 3860
5766 : 3861 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5767 : 3862 RD_DATA = .MLD1; !READ AND SAVE THE REGISTER
5768 : 3863
5769 : 3864 if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE; !READ MLD1 FOR WRT_MASK
5770 : 3865
5771 : 3866 !SET ERR_FLG IF NEQ
5772 : 3867 CLR_MBUS; !CLEAR THE MBUS
5773 : 3868 end;
5777 :

```

```

5781 021654 004167 162164 RD.D1: JSR R1,$SAVE2 ; 3795
5782 021660 005076 000010 CLR @10(SP) ; ERR.FLG 3848
5783 021664 152777 000010 171402 BISB #10,@ML.REG+120 ; 3849
5784 021672 142777 000077 171254 BICB #77,@ML.REG ; 3850
5785 021700 152777 000061 171246 BISB #61,@ML.REG
5786 021706 152777 000020 171360 BISB #20,@ML.REG+120 ; 3851
5787 021714 026727 171216 000001 CMP REG.INIT.FLG,#1 ; 3853
5788 021722 001017 BNE 1$ ;
5789 021724 152777 000040 171262 BISB #40,@ML.REG+40 ; 3856
5790 021732 016701 171644 MOV ML,DUT,R1
5791 021736 042701 177770 BIC #177770,R1
5792 021742 142777 000007 171244 BICB #7,@ML.REG+40
5793 021750 150177 171240 BISB R1,@ML.REG+40
5794 021754 152777 000010 171312 BISB #10,@ML.REG+120 ; 3858
5795 021762 016600 000012 1$: MOV 12(SP),R0 ; INDEX,* 3861
5796 021766 006300 ASL R0
5797 021770 006300 ASL R0
5798 021772 006300 ASL R0
5799 021774 010001 MOV R0,R1
5800 021776 016100 013156 MOV ML,REG+2(R1),R0
5801 022002 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5802 022006 046100 013160 BIC ML,REG+4(R1),R0

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

```
5804          ;ML4
5805          ;
5806
5807 022012 016102 013162          MOV    ML.REG+6(R1),R2
5808 022016 050002          BIS    R0,R2
5809 022020 010267 171102          MOV    R2,WT.DATA
5810 022024 017767 171314 171076  MOV    @ML.REG+170,RD.DATA          ;
5811 022032 026767 171072 171066  CMP    RD.DATA,WT.DATA          ;
5812 022040 001403          BEQ    2$
5813 022042 012776 000001 000010  MOV    #1,@10(SP)          ; *,ERR.FLG
5814 022050 152777 000040 171136 2$:  BISB  #40,@ML.REG+40
5815 022056 016702 171520          MOV    ML.DUT,R2
5816 022062 042702 177770          BIC    #177770,R2
5817 022066 142777 000007 171120  BICB  #7,@ML.REG+40
5818 022074 150277 171114          BISB  R2,@ML.REG+40
5819 022100 000207          RTS    PC          ;
5820
5821          ; Routine Size: 75 words
5822          ; Maximum stack depth per invocation: 3 words
5827
5828
```

3862
3864

3795

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (41)

5830 :ML4
5831 :
5832 :
5833 :
5834 :
5835 :
5836 :
5837 :
5838 :
5839 :
5840 :
5841 :
5842 :
5843 :
5844 :
5845 :
5846 :
5847 :
5848 :
5849 :
5850 :
5854 :
5858 :
5859 :
5860 :
5861 :
5862 :
5863 :
5864 :
5865 :
5866 :
5867 :
5868 :
5869 :
5870 :
5871 :
5872 :
5873 :
5874 :
5875 :
5880 :
5881 :

3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886

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;

MLD2 = WRT_MASK;

DAT_DM = ZERO;

end;

!SET DATA DIAG MODE

!LOAD WRT_D2 WITH GENERATED WRT_MASK PATTERN

!CLEAR DATA DIAG MODE

```

WRT.D2: JSR      R1,SSAVE2          ;
        BISB   #10,2ML.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   R2,2ML.REG+200
        BICB  #10,2ML.REG+120    ;
        RTS   PC                  ;
    
```

3869
3883
3884
3885
3869

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (42)

5883 :ML4
5884 :
5885 :
5886 :
5887 :
5888 :
5889 :
5890 :
5891 :
5892 :
5893 :
5894 :
5895 :
5896 :
5897 :
5898 :
5899 :
5900 :
5901 :
5902 :
5903 :
5904 :
5905 :
5906 :
5907 :
5908 :
5909 :
5910 :
5911 :
5912 :
5913 :
5914 :
5915 :
5916 :
5917 :
5918 :
5919 :
5920 :
5921 :
5922 :
5923 :
5924 :
5925 :
5926 :
5927 :
5928 :
5929 :
5930 :
5931 :
5932 :
5933 :
5934 :
5935 :
5936 :
5937 :

3187
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938

routine RD_D2 (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARE THE CONTENTS OF THE
DATA DIAG REGISTER 2
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 ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

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.

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-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (42)

```

5939 :ML4
5940 :
5941 :
5942 : 3939 .ERR_FLG = ZERO; !CLEAR ERROR FLAG
5943 : 3940 DAT_DM = ONE; !SET DATA DIAG MODE
5944 : 3941 ML_FUNC = write; !LOAD WRITE FUNC TO CS1
5945 : 3942 DAT_CLK = ONE; !DO A DATA CLOCK
5946 : 3943
5947 : 3944 if .REG_INIT_FLG IS_SET !SEE IF CALLER IS REG INIT TEST
5948 : 3945 then !CLEAR MBUS TO GENERATE INIT DATA
5949 : 3946 begin
5950 : 3947 CLR_MBUS;
5951 : 3948 DAT_DM = ONE;
5952 : 3949 end;
5953 : 3950
5954 : 3951 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5955 : 3952 RD_DATA = .MLD2; !READ AND SAVE THE REGISTER
5956 : 3953
5957 : 3954 if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE; !READ MLD2 FR WRT_MASK PATTERN
5958 : 3955
5959 : 3956 !SET ERROR FLAG IF NEQ
5960 : 3957 CLR_MBUS; !CLR MASS BUSS
5961 : 3958 end;
5965 :
5969 022166 004167 161652 RD.D2: JSR R1,$SAVE2 ; 3887
5970 022172 005076 000010 CLR @10(SP) ; ERR.FLG 3939
5971 022176 152777 000010 171070 BISB #10,@ML.REG+120 ; 3940
5972 022204 142777 000077 170742 BICB #77,@ML.REG ; 3941
5973 022212 152777 000061 170734 BISB #61,@ML.REG ;
5974 022220 152777 000020 171046 BISB #20,@ML.REG+120 ; 3942
5975 022226 026727 170704 000001 CMP REG.INIT.FLG,#1 ; 3944
5976 022234 001017 BNE 1$ ;
5977 022236 152777 000040 170750 BISB #40,@ML.REG+40 ; 3946
5978 022244 016701 171332 MOV ML.DUT,R1 ;
5979 022250 042701 177770 BIC #177770,R1 ;
5980 022254 142777 000007 170732 BICB #7,@ML.REG+40 ;
5981 022262 150177 170726 BISB R1,@ML.REG+40 ;
5982 022266 152777 000010 171000 BISB #10,@ML.REG+120 ; 3948
5983 022274 016600 000012 1$: MOV 12(SP),R0 ; INDEX,* 3951
5984 022300 006300 ASL R0 ;
5985 022302 006300 ASL R0 ;
5986 022304 006300 ASL R0 ;
5987 022306 010001 MOV R0,R1 ;
5988 022310 016100 013156 MOV ML.REG+2(R1),R0 ;
5989 022314 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5990 022320 046100 013160 BIC ML.REG+4(R1),R0 ;
5991 022324 016102 013162 MOV ML.REG+6(R1),R2 ;
5992 022330 050002 BIS R0,R2 ;

```

CZMLAB0 ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 162

SEQ 0149

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

5994      ;ML4
5995      :
5996
5997 022332 010267 170570      MOV      R2,WT.DATA
5998 022336 017767 171012 170564  MOV      @ML.REG+200,RD.DATA      :
5999 022344 026767 170560 170554  CMP      RD.DATA,WI.DATA      :
6000 022352 001403          BEQ      2$                      :
6001 022354 012776 000001 000010  MOV      #1,@10(SP)              : *.ERR.FLG
6002 022362 152777 000040 170624 2$:  BISB    #40,@ML.REG+40
6003 022370 016702 171206      MOV      ML.DUT,R2
6004 022374 042702 177770      BIC     #177770,R2
6005 022400 142777 000007 170606  BICB    #7,@ML.REG+40
6006 022406 150277 170602      BISB    R2,@ML.REG+40
6007 022412 000207          RTS      PC                      :

```

3952
3954

3887

: Routine Size: 75 words
: Maximum stack depth per invocation: 3 words

6008
6009
6010
6015
6016

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (43)

```

6018 :ML4
6019 :
6020 :
6021 : 3959 routine WRT_D3 (TST_PAT, index) : novalue =
6022 : 3960 begin
6023 : 3961
6024 : 3962 |++
6025 : 3963 | FUNCTIONAL DESCRIPTION:
6026 : 3964 |
6027 : 3965 | LOADS THE DATA DIAG
6028 : 3966 | REGISTER 3 WITH A DATA PATTERN
6029 : 3967 | GENERATED BY THE MACRO
6030 : 3968 | WRT_MASK.
6031 : 3969 |
6032 : 3970 | FORMAL PARAMETERS:
6033 : 3971 |
6034 : 3972 | TST PAT
6035 : 3973 | CURRENT DATA PATTERN TO BE
6036 : 3974 | LOADED IN THE REGISTER.
6037 : 3975 |
6038 : 3976 | INDEX
6039 : 3977 | USE BY THE MACRO WRT MASK
6040 : 3978 | TO SELECT THE CURRENT REGISTERS
6041 : 3979 | ADDRESS, FORCED HI, FORCED
6042 : 3980 | LO AND DON'T CARE MASK
6043 : 3981 | INFORMATION.
6044 : 3982 |
6045 : 3983 |
6046 : 3984 |
6047 : 3985 | DAT_DM = ONE;
6048 : 3986 | MLE2_MASK = %0'000377';
6049 : 3987 | MLE2 = WRT_MASK;
6050 : 3988 | MLE2_MASK = %0'100300';
6051 : 3989 | DAT_DM = ZERO;
6052 : 3990 end;

```

```

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

```

```

CJ56
6060 022414 004167 161424 WRT.D3: JSR R1,$SAVE2 : 3959
6061 022420 152777 000010 170646 BISB #10,2ML.REG+120 : 3985
6062 022426 012767 000377 170706 MOV #377,ML.REG+166 : 3986
6063 022434 016600 000010 MOV 10(SP),R0 : INDEX,* 3987
6064 022440 006300 ASL R0
6065 022442 006300 ASL R0
6066 022444 006300 ASL R0
6067 022446 010001 MOV R0,R1
6068 022450 016100 013156 MOV ML.REG+2(R1),R0
6069 022454 056600 000012 BIS 12(SP),R0 ; TST.PAT,*
6070 022460 046100 013160 BIC ML.REG+4(R1),R0
6071 022464 016102 013162 MOV ML.REG+6(R1),R2

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

6073
6074
6075
6076 022470 050002
6077 022472 010277 170636
6078 022476 012767 100300 170636
6079 022504 142777 000010 170562
6080 022512 000207
6081
6082
6083
6088
6089

:ML4
:

BIS R0,R2
MOV R2,@ML.REG+160
MOV #-77500,ML.REG+166
BICB #10,@ML.REG+120
RTS PC

:
:
:

3988
3989
3959

: Routine Size: 32 words
: Maximum stack depth per invocation: 3 words

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (44)

```
6091 :ML4
6092 :
6093 :
6094 : 3991 routine RD_D3 (TST_PAT, index, ERR_FLG) : novalue =
6095 : 3992     begin
6096 : 3993
6097 : 3994     +-
6098 : 3995     FUNCTIONAL DESCRIPTION:
6099 : 3996
6100 : 3997         COMPARES THE CONTENTS OF THE
6101 : 3998         DATA DIAG REGISTER 3
6102 : 3999         WITH THE MASKED DATA PATTERN
6103 : 4000         GENERATED BY THE MACRO 'WRT_MASK'.
6104 : 4001
6105 : 4002         IF THE COMPARE IS NOT EQUAL THE
6106 : 4003         FORMAL PARAMETER 'ERR_FLG' IS
6107 : 4004         ASSIGNED A ONE TO INDICATE THE
6108 : 4005         ERROR.
6109 : 4006
6110 : 4007     FORMAL PARAMETERS:
6111 : 4008
6112 : 4009         TST PAT
6113 : 4010         DATA PATTERN TO BE MASKED AND
6114 : 4011         COMPARED AGAINST THE CONTENTS
6115 : 4012         OF THE REGISTER UNDER TEST.
6116 : 4013
6117 : 4014         INDEX
6118 : 4015         USED BY THE MACRO WRT_MASK TO
6119 : 4016         SELECT THE CURRENT REGISTER ADDRESS,
6120 : 4017         FORCED HI, FORCED LO AND DON'T CARE
6121 : 4018         MASK INFORMATION
6122 : 4019
6123 : 4020         ERR_FLG
6124 : 4021         CONTAINS THE ADDRESS (PASSED BY REF)
6125 : 4022         OF THE CALLERS ERROR_FLG TO ENABLE THE
6126 : 4023         CALLER TO EXAMINE THE ERROR STATUS
6127 : 4024         OF THE ROUTINE CALL.
6128 : 4025
6129 : 4026     IMPLICIT INPUTS:
6130 : 4027         WT DATA
6131 : 4028         GETS LOADED WITH THE GENERATED
6132 : 4029         WRT_MASK DATA PATTERN THUS ALLOWING
6133 : 4030         CALLER TO PRINT FAILING GOOD DATA.
6134 : 4031
6135 : 4032         RD DATA
6136 : 4033         GETS LOADED WITH DATA READ FROM THE
6137 : 4034         REGISTER THUS ALLOWING CALLER
6138 : 4035         TO PRINT FAILING BAD DATA.
6139 : 4036
6140 : 4037     IMPLICIT OUTPUTS:
6141 : 4038         GLOBAL LOCATION WR_DATA
6142 : 4039         AND RD_DATA LOADED WITH GOOD
6143 : 4040         AND BAD REGISTER DATA
6144 : 4041
6145 : 4042     --
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (44)

```

6147 :ML4
6148 :
6149 :
6150 :      4043
6151 :      4044      .ERR_FLG = ZERO;          !CLEAR ERROR FLAG
6152 :      4045      MLE2_MASK = %o'000377';    !SET DATA DIAG MODE
6153 :      4046      DAT_DM = ONE;
6154 :      4047      ML_FUNC = write;          !LOAD WRITE FUNCTION TO MLCS1
6155 :      4048      DAT_CLK = ONE;          !DO A DATA CLOCK
6156 :      4049
6157 :      4050      if .REG_INIT_FLG IS_SET
6158 :      4051      then
6159 :      4052          begin
6160 :      4053          CLR_MBUS;
6161 :      4054          DAT_DM = ONE;
6162 :      4055          end;
6163 :      4056
6164 :      4057      WT_DATA = WRT_MASK;          !SAVE THE DATA WRITTEN TO THE REGISTER
6165 :      4058      RD_DATA = .MLE2 or .IGNORE; !READ AND SAVE THE REGISTER
6166 :      4059
6167 :      4060      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;    !READ THE REG FOR WRT_MASK
6168 :      4061
6169 :      4062      MLE2_MASK = %o'100300';
6170 :      4063      CLR_MBUS;          !CLEAR THE MASS BUS
6171 :      4064      end;
6175 :

```

```

6179 022514 004167 161324      RD.D3: JSR      R1,$SAVE2          ;
6180 022520 005076 000010      CLR      @10(SP)          ; ERR.FLG
6181 022524 012767 000377 170610      MOV      #377,ML.REG+166 ;
6182 022532 152777 000010 170534      BISB    #10,@ML.REG+120 ;
6183 022540 142777 000077 170406      BICB    #77,@ML.REG      ;
6184 022546 152777 000061 170400      BISB    #61,@ML.REG      ;
6185 022554 152777 000020 170512      BISB    #20,@ML.REG+120 ;
6186 022562 026727 170350 000001      CMP     REG.INIT.FLG,#1  ;
6187 022570 001017      BNE     1$              ;
6188 022572 152777 000040 170414      BISB    #40,@ML.REG+40  ;
6189 022600 016701 170776      MOV     ML.DUT,R1
6190 022604 042701 177770      BIC     #177770,R1
6191 022610 142777 000007 170376      BICB    #7,@ML.REG+40
6192 022616 150177 170372      BISB    R1,@ML.REG+40
6193 022622 152777 000010 170444      BISB    #10,@ML.REG+120 ;
6194 022630 016600 000012      1$:  MOV     12(SP),R0      ; INDEX,*
6195 022634 006300      ASL     R0
6196 022636 006300      ASL     R0
6197 022640 006300      ASL     R0
6198 022642 010001      MOV     R0,R1
6199 022644 016100 013156      MOV     ML.REG+2(R1),R0
6200 022650 056600 000014      BIS     14(SP),R0      ; TST.PAT,*

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

6202          ;ML4
6203          ;
6204
6205 022654 046100 013160          BIC    ML.REG+4(R1),R0
6206 022660 016102 013162          MOV    ML.REG+6(R1),R2
6207 022664 050002                   BIS    R0,R2
6208 022666 010267 170234          MOV    R2,WT.DATA
6209 022672 017702 170436          MOV    @ML.REG+160,R2          ;
6210 022676 056102 013162          BIS    ML.REG+6(R1),R2          ;
6211 022702 010267 170222          MOV    R2,PD.DATA          ;
6212 022706 026767 170214 170214  CMP    WT.DATA,RD.DATA          ;
6213 022714 001403                   BEQ    2$
6214 022716 012776 000001 000010  MOV    #1,@10(SP)          ; *,ERR.FLG
6215 022724 012767 100300 170410 2$: MOV    #-77500,ML.REG+166          ;
6216 022732 152777 000040 170254  BISB   #40,@ML.REG+40          ;
6217 022740 016702 170636          MOV    ML.DUT,R2
6218 022744 042702 177770          BIC    #177770,R2
6219 022750 142777 000007 170236  BICB   #7,@ML.REG+40
6220 022756 150277 170232          BISB   R2,@ML.REG+40
6221 022762 000207                   RTS    PC          ;
6222
6223          ; Routine Size: 84 words
6224          ; Maximum stack depth per invocation: 3 words
6229
6230

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (45)

```
6232 :ML4
6233 :
6234 :
6235 : 4065 routine WRT_DS (TST_PAT, index) : novalue =
6236 : 4066     begin
6237 : 4067
6238 : 4068     !++
6239 : 4069     FUNCTIONAL DESCRIPTION:
6240 : 4070     DUMMY ROUTINE CALL TO ASSIST IN THE READ
6241 : 4071     WRITE REGISTER ALGORITHM
6242 : 4072
6243 : 4073     FORMAL PARAMETERS:
6244 : 4074     TST_PAT
6245 : 4075     DATA PATTERN TO BE MASKED AND
6246 : 4076     COMPARED AGAINST THE CONTENTS
6247 : 4077     OF THE REGISTER UNDER TEST
6248 : 4078
6249 : 4079     INDEX
6250 : 4080     USED BY THE MACRO WRT_MASK TO
6251 : 4081     SELECT THE CURRENT REGISTER ADDRESS,
6252 : 4082     FORCED HI, FORCED LO AND DON'T CARE
6253 : 4083     MASK INFORMATION.
6254 : 4084
6255 : 4085     --
6256 : 4086
6257 : 4087
6258 : 4088     DRIVE STATUS REG IS READ ONLY
6259 : 4089     return;
6260 : 4090     end;
6261 :
6262 :
6263 :
6264 :
6265 :
6266 :
6267 :
6268 : 022764 000207 WRT_DS: RTS PC ;
6269 :
6270 : ; Routine Size: 1 word
6271 : ; Maximum stack depth per invocation: 0 words
6272 :
6273 :
6274 :
6275 :
6276 :
6277 :
```

4065

6279 :ML4
6280 :
6281 :
6282 :
6283 :
6284 :
6285 :
6286 :
6287 :
6288 :
6289 :
6290 :
6291 :
6292 :
6293 :
6294 :
6295 :
6296 :
6297 :
6298 :
6299 :
6300 :
6301 :
6302 :
6303 :
6304 :
6305 :
6306 :
6307 :
6308 :
6309 :
6310 :
6311 :
6312 :
6313 :
6314 :
6315 :
6316 :
6317 :
6318 :
6319 :
6320 :
6321 :
6322 :
6323 :
6324 :
6325 :
6326 :
6327 :
6328 :
6329 :
6330 :
6331 :
6332 :
6333 :

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (46)

```
4091 routine RD_DS (TST_PAT, index, ERR_FLG) : novalue =
4092   begin
4093
4094   !++
4095   FUNCTIONAL DESCRIPTION:
4096
4097       COMPARES THE CONTENTS OF THE
4098       DRIVE STATUS REGISTER WITH THE
4099       MASKED DATA PATTERN
4100       GENERATED BY THE MACRO 'WRT_MASK'.
4101
4102       IF THE COMPARE IS NOT EQUAL THEN
4103       THE FORMAL PARAMETER 'ERR_FLG' IS
4104       ASSIGNED A ONE TO INDICATE THE
4105       ERROR.
4106
4107   FORMAL PARAMETERS:
4108
4109       TST_PAT
4110       DATA PATTERN TO BE MASKED AND
4111       COMPARED AGAINST THE CONTENTS
4112       OF THE REGISTER UNDER TEST.
4113
4114       INDEX
4115       USED BY THE MACRO WRT_MASK TO
4116       SELECT THE CURRENT REGISTER ADDRESS,
4117       FORCED HI, FORCED LO AND DON'T CARE
4118       MASK INFORMATION.
4119
4120       ERR_FLG
4121       CONTAINS THE ADDRESS (PASSED BY REF)
4122       OF THE CALLERS ERROR FLG TO ENABLE THE
4123       CALLER TO EXAMINE THE ERROR STATUS
4124       OF THE ROUTINE CALL.
4125
4126   IMPLICIT INPUTS:
4127       WT_DATA
4128       GETS LOADED WITH THE GENERATED
4129       WRT_MASK DATA PATTERN THUS ALLOWING
4130       CALLER TO PRINT FAILING GOOD DATA.
4131
4132       RD_DATA
4133       GETS LOADED WITH DATA READ FROM THE
4134       REGISTER THUS ALLOWING CALLER
4135       TO PRINT FAILING BAD DATA.
4136
4137   IMPLICIT OUTPUTS:
4138       GLOBAL LOCATION WR_DATA
4139       AND RD_DATA LOADED WITH GOOD
4140       AND BAD REGISTER DATA
4141
4142   --
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (46)

```

6335 :ML4
6336 :
6337 :
6338 :      4143
6339 :      4144      .ERR_FLG = ZERO;
6340 :      4145      WT_DATA = WRT_MASK;
6341 :      4146      RD_DATA = .MLDS or .IGNORE;
6342 :      4147
6343 :      4148      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;
6344 :      4149
6345 :      4150
6346 :      4151      end;
6350 :
6354 022766 004167 161052      RD.DS: JSR      R1,$SAVE2
6355 022772 005076 000010      CLR      @10(SP)
6356 022776 016600 000012      MOV      12(SP),R0
6357 023002 006300      ASL      R0
6358 023004 006300      ASL      R0
6359 023006 006300      ASL      R0
6360 023010 010001      MOV      R0,R1
6361 023012 016100 013156      MOV      ML.REG+2(R1),R0
6362 023016 056600 000014      BIS      14(SP),R0
6363 023022 046100 013160      BIC      ML.REG+4(R1),R0
6364 023026 016102 013162      MOV      ML.REG+6(R1),R2
6365 023032 050002      BIS      R0,R2
6366 023034 010267 170066      MOV      R2,WT_DATA
6367 023040 017702 170160      MOV      @ML.REG+50,R2
6368 023044 056102 013162      BIS      ML.REG+6(R1),R2
6369 023050 010267 170054      MOV      R2,RD_DATA
6370 023054 026767 170046 170046      CMP      WT_DATA,RD_DATA
6371 023062 001403      BEQ      JS
6372 023064 012776 000001 000010      MOV      #1,@10(SP)
6373 023072 000207      1$: RTS      PC
6374 :
6375 :
6376 :
6381 :
6382 :

```

```

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

```

```

4091
4144
4145
4146
4148
4091

```

```

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

```


22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (47)

6384 :ML4
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 :
6431 :
6432 :
6433 :
6434 :
6435 :
6436 :
6437 :
6438 :

```

4152 routine WRT_REG (TST_PAT, REG_SEL, index) : novalue =
4153 begin
4154
4155 !++
4156 !FUNCTIONAL DESCRIPTION:
4157 !A ROUTINE TO SELECTIVELY
4158 !CALLED ROUTINES WHICH
4159 !WRITE TO INDIVIDUAL ML11
4160 !REGISTERS
4161
4162 !FORMAL PARAMETERS:
4163 !REG_SEL
4164 !CASE SELECT EXPRESSION TO
4165 !SELECT THE WRITE REGISTER
4166 !ROUTINE TO CALLED
4167
4168 !TST_PAT
4169 !DATA PATTERN WHICH THE SELECTED
4170 !REGISTER WILL BE TESTED AGAINST
4171
4172 !INDEX
4173 !LOADED WITH THE ML_REG INDEX
4174 !SELECT NUMBER OF THE REGISTER
4175 !BEING TESTED
4176
4177 !SIDE EFFECTS:
4178 !WHEN A WRITE REGISTER ROUTINE IS CALLED
4179 !THE VARIABLE 'INDEX' FROM THE CALLING
4180 !TEST IS LOADED WITH THE REGISTERS
4181 !ML_REG INDEX NUMBER.
4182
4183 !THIS ENABLES THE CALLING TEST TO FIND
4184 !THE FAILING REGISTER ADDRESS.
4185
4186
4187
4188 case .REG_SEL from 0 to 13 of
4189 set
4190
4191 [0] :
4192     WRT_CS1 (.TST_PAT, .index = 0);
4193
4194 [1] :
4195     WRT_ER (.TST_PAT, .index = 6);
4196
4197 [2] :
4198     WRT_DA (.TST_PAT, .index = 3);
4199
4200 [3] :
4201     WRT_MR (.TST_PAT, .index = 10);
4202
4203 [4] :

```

```

!SELECT THE WRITE REGISTER ROUTINE CALL
!CALL ROUTINE TO LOAD MLCS1
!CALL ROUTINE TO LOAD MLER
!CALL ROUTINE TO LOAD MLDA
!CALL ROUTINE TO LOAD MLMR

```

6440 :ML4
6441 :
6442 :
6443 :
6444 :
6445 :
6446 :
6447 :
6448 :
6449 :
6450 :
6451 :
6452 :
6453 :
6454 :
6455 :
6456 :
6457 :
6458 :
6459 :
6460 :
6461 :
6462 :
6463 :
6464 :
6465 :
6466 :
6467 :
6468 :
6469 :
6470 :
6471 :
6472 :
6473 :
6477 :

4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234

end;

WRT_E1 (.TST_PAT, .index = 13); !CALL ROUTINE TO LOAD MLE1
[5] :
WRT_E2 (.TST_PAT, .index = 14); !CALL ROUTINE TO LOAD MLE2
[6] :
WRT_PA (.TST_PAT, .index = 8); !CALL ROUTINE TO LOAD MLPA
[7] :
WRT_PD (.TST_PAT, .index = 19); !CALL ROUTINE TO LOAD MLPD
[8] :
WRT_EE (.TST_PAT, .index = 17); !CALL ROUTINE TO LOAD MLEE
[9] :
WRT_EL (.TST_PAT, .index = 18); !CALL ROUTINE TO LOAD MLEL
[10] :
WRT_DS (.TST_PAT, .index = 5); !CALL ROUTINE TO LOAD MLDS
[11] :
WRT_D1 (.TST_PAT, .index = 15); !CALL ROUTINE TO LOAD MLD1
[12] :
WRT_D2 (.TST_PAT, .index = 16); !CALL ROUTINE TO LOAD MLD2
[13] :
WRT_D3 (.TST_PAT, .index = 14); !CALL ROUTINE TO LOAD MLE2
tes;

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (47)

6481 023074 004167 160744
6482 023100 016600 000010
6483 023104 016601 000014
6484 023110 016602 000012
6485 023114 006302
6486 023116 066207 023122
6487 023122 000034
6488 023124 000050
6489 023126 000066
6490 023130 000104
6491 023132 000122
6492 023134 000140
6493 023136 000156

WRT.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\$-1\$
.WORD 3\$-1\$
.WORD 4\$-1\$
.WORD 5\$-1\$
.WORD 6\$-1\$
.WORD 7\$-1\$
.WORD 8\$-1\$

:
: INDEX,*
: TST.PAT,*
: REG.SEL,*

4152
4192
4188

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

6495							
6496			:ML4				
6497			:				
6498	023140	000174		.WORD	9\$-1\$		
6499	023142	000212		.WORD	10\$-1\$		
6500	023144	000230		.WORD	11\$-1\$		
6501	023146	000246		.WORD	12\$-1\$		
6502	023150	000264		.WORD	13\$-1\$		
6503	023152	000302		.WORD	14\$-1\$		
6504	023154	000320		.WORD	15\$-1\$		
6505	023156	010146	2\$:	MOV	R1, -(SP)	:	4192
6506	023160	005010		CLR	(R0)	:	
6507	023162	005046		CLR	-(SP)	:	
6508	023164	004767	174272	JSR	PC, WRT.CS1	:	
6509	023170	000532		BR	16\$:	4188
6510	023172	010146	3\$:	MOV	R1, -(SP)	:	4195
6511	023174	012710	000006	MOV	#6, (R0)	:	
6512	023200	011046		MOV	(R0), -(SP)	:	
6513	023202	004767	174432	JSR	PC, WRT.ER	:	
6514	023206	000523		BR	16\$:	4188
6515	023210	010146	4\$:	MOV	R1, -(SP)	:	4198
6516	023212	012710	000003	MOV	#3, (R0)	:	
6517	023216	011046		MOV	(R0), -(SP)	:	
6518	023220	004767	174572	JSR	PC, WRT.DA	:	
6519	023224	000514		BR	16\$:	4188
6520	023226	010146	5\$:	MOV	R1, -(SP)	:	4201
6521	023230	012710	000012	MOV	#12, (R0)	:	
6522	023234	011046		MOV	(R0), -(SP)	:	
6523	023236	004767	174732	JSR	PC, WRT.MR	:	
6524	023242	000505		BR	16\$:	4188
6525	023244	010146	6\$:	MOV	R1, -(SP)	:	4204
6526	023246	012710	000015	MOV	#15, (R0)	:	
6527	023252	011046		MOV	(R0), -(SP)	:	
6528	023254	004767	175300	JSR	PC, WRT.E1	:	
6529	023260	000476		BR	16\$:	4188
6530	023262	010146	7\$:	MOV	R1, -(SP)	:	4207
6531	023264	012710	000016	MOV	#16, (R0)	:	
6532	023270	011046		MOV	(R0), -(SP)	:	
6533	023272	004767	175470	JSR	PC, WRT.E2	:	
6534	023276	000467		BR	16\$:	4188
6535	023300	010146	8\$:	MOV	R1, -(SP)	:	4210
6536	023302	012710	000010	MOV	#10, (R0)	:	
6537	023306	011046		MOV	(R0), -(SP)	:	
6538	023310	004767	175036	JSR	PC, WRT.PA	:	
6539	023314	000460		BR	16\$:	4188
6540	023316	010146	9\$:	MOV	R1, -(SP)	:	4213
6541	023320	012710	000023	MOV	#23, (R0)	:	
6542	023324	011046		MOV	(R0), -(SP)	:	
6543	023326	004767	175666	JSR	PC, WRT.PD	:	
6544	023332	000451		BR	16\$:	4188
6545	023334	010146	10\$:	MOV	R1, -(SP)	:	4216
6546	023336	012710	000021	MOV	#21, (R0)	:	
6547	023342	011046		MOV	(R0), -(SP)	:	
6548	023344	004767	176156	JSR	PC, WRT.EE	:	
6549	023350	000442		BR	16\$:	4188

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

6551      ;ML4
6552      ;
6553
6554 023352 010146      11$:  MOV    R1,-(SP)      ;
6555 023354 012710 000022  MOV    #22,(R0)      ;
6556 023360 011046      MOV    (R0),-(SP)    ;
6557 023362 004767 176076  JSR    PC,WRT.EL     ;
6558 023366 000433      BR     16$           ;
6559 023370 010146      12$:  MOV    R1,-(SP)      ;
6560 023372 012710 000005  MOV    #5,(R0)       ;
6561 023376 011046      MOV    (R0),-(SP)    ;
6562 023400 004767 177360  JSR    PC,WRT.DS     ;
6563 023404 000424      BR     16$           ;
6564 023406 010146      13$:  MOV    R1,-(SP)      ;
6565 023410 012710 000017  MOV    #17,(R0)      ;
6566 023414 011046      MOV    (R0),-(SP)    ;
6567 023416 004767 176146  JSR    PC,WRT.D1     ;
6568 023422 000415      BR     16$           ;
6569 023424 010146      14$:  MOV    R1,-(SP)      ;
6570 023426 012710 000020  MOV    #20,(R0)      ;
6571 023432 011046      MOV    (R0),-(SP)    ;
6572 023434 004767 176442  JSR    PC,WRT.D2     ;
6573 023440 000406      BR     16$           ;
6574 023442 010146      15$:  MOV    R1,-(SP)      ;
6575 023444 012710 000016  MOV    #16,(R0)      ;
6576 023450 011046      MOV    (R0),-(SP)    ;
6577 023452 004767 176736  JSR    PC,WRT.D3     ;
6578 023456 022626      16$:  CMP    (SP)+,(SP)+   ;
6579 023460 000207      RTS    PC            ;
6580
6581
6582
6587
6588

```

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

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (48)

6590 :ML4
6591 :
6592 :
6593 :
6594 :
6595 :
6596 :
6597 :
6598 :
6599 :
6600 :
6601 :
6602 :
6603 :
6604 :
6605 :
6606 :
6607 :
6608 :
6609 :
6610 :
6611 :
6612 :
6613 :
6614 :
6615 :
6616 :
6617 :
6618 :
6619 :
6620 :
6621 :
6622 :
6623 :
6624 :
6625 :
6626 :
6627 :
6628 :
6629 :
6630 :
6631 :
6632 :
6633 :
6634 :
6635 :
6636 :
6637 :
6638 :
6639 :
6640 :
6641 :
6642 :
6643 :
6644 :

```

4235 routine RD_REG (TST_PAT, REG_SEL, ERR_FLG) : novalue =
4236     begin
4237
4238     !++
4239     FUNCTIONAL DESCRIPTION:
4240     A ROUTINE TO SELECTIVELY
4241     CALLED ROUTINES WHICH
4242     READ TO INDIVIDUAL ML11
4243     REGISTERS.
4244
4245     FORMAL PARAMETERS:
4246     ERR_FLG
4247     CONTAINS THE ADDRESS (PASSED BY REF)
4248     OF THE CALLERS ERROR FLG TO ENABLE THE
4249     CALLER TO EXAMINE THE ERROR STATUS
4250     OF THE ROUTINE CALL.
4251
4252     REG_SEL
4253     CASE SELECT EXPRESSION TO
4254     SELECT THE WRITE REGISTER
4255     ROUTINE TO CALLED
4256
4257     TST_PAT
4258     DATA PATTERN WHICH THE SELECTED
4259     REGISTER WILL BE TESTED AGAINST
4260     !--
4261
4262     case .REG_SEL from 0 to 13 of
4263     set                                     !SELECT THE READ REGISTER ROUTINE CALL
4264
4265     [0] :
4266         RD_CS1 (.TST_PAT, 0, .ERR_FLG);    !CALL ROUTINE TO READ MLCS1
4267
4268     [1] :
4269         RD_ER (.TST_PAT, 6, .ERR_FLG);    !CALL ROUTINE TO READ MLER
4270
4271     [2] :
4272         RD_DA (.TST_PAT, 3, .ERR_FLG);    !CALL ROUTINE TO READ MLDA
4273
4274     [3] :
4275         RD_MR (.TST_PAT, 10, .ERR_FLG);   !CALL ROUTINE TO READ MLMR
4276
4277     [4] :
4278         RD_E1 (.TST_PAT, 13, .ERR_FLG);   !CALL ROUTINE TO READ MLE1
4279
4280     [5] :
4281         RD_E2 (.TST_PAT, 14, .ERR_FLG);   !CALL ROUTINE TO READ MLE2
4282
4283     [6] :
4284         RD_PA (.TST_PAT, 8, .ERR_FLG);    !CALL ROUTINE TO READ MLPA
4285
4286

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (48)

6646 :ML4
6647 :
6648 :
6649 : 4287
6650 : 4288
6651 : 4289
6652 : 4290
6653 : 4291
6654 : 4292
6655 : 4293
6656 : 4294
6657 : 4295
6658 : 4296
6659 : 4297
6660 : 4298
6661 : 4299
6662 : 4300
6663 : 4301
6664 : 4302
6665 : 4303
6666 : 4304
6667 : 4305
6668 : 4306
6669 : 4307
6670 : 4308
6671 : 4309

[7] : RD_PD (.TST_PAT, 19, .ERR_FLG); !CALL ROUTINE TO READ MLPD
[8] : RD_EE (.TST_PAT, 17, .ERR_FLG); !CALL ROUTINE TO READ MLEE
[9] : RD_EL (.TST_PAT, 18, .ERR_FLG); !CALL ROUTINE TO READ MLEL
[10] : RD_DS (.TST_PAT, 5, .ERR_FLG); !CALL ROUTINE TO READ MLDS
[11] : RD_D1 (.TST_PAT, 15, .ERR_FLG); !CALL ROUTINE TO READ MLD1
[12] : RD_D2 (.TST_PAT, 16, .ERR_FLG); !CALL ROUTINE TO READ MLD2
[13] : RD_D3 (.TST_PAT, 14, .ERR_FLG); !CALL ROUTINE TO READ MLE2
tes:

end;

6679 023462 004167 160356
6680 023466 016600 000010
6681 023472 016601 000014
6682 023476 016602 000012
6683 023502 006302
6684 023504 066207 023510
6685 023510 000034
6686 023512 000050
6687 023514 000066
6688 023516 000104
6689 023520 000122
6690 023522 000140
6691 023524 000156
6692 023526 000174
6693 023530 000212
6694 023532 000230
6695 023534 000246
6696 023536 000264
6697 023540 000302
6698 023542 000320
6699 023544 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\$-1\$
.WORD 3\$-1\$
.WORD 4\$-1\$
.WORD 5\$-1\$
.WORD 6\$-1\$
.WORD 7\$-1\$
.WORD 8\$-1\$
.WORD 9\$-1\$
.WORD 10\$-1\$
.WORD 11\$-1\$
.WORD 12\$-1\$
.WORD 13\$-1\$
.WORD 14\$-1\$
.WORD 15\$-1\$
2\$: MOV R1,-(SP)

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

4235
4267
4263

4267

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Address
6701							
6702							
6703							
6704	023546	005046			CLR	-(SP)	
6705	023550	010046			MOV	R0,-(SP)	
6706	023552	004767	173754		JSR	PC,RD.CS1	
6707	023556	000532			BR	16\$	4263
6708	023560	010146		3\$:	MOV	R1,-(SP)	4270
6709	023562	012746	000006		MOV	#6,-(SP)	
6710	023566	010046			MOV	R0,-(SP)	
6711	023570	004767	174114		JSR	PC,RD.ER	
6712	023574	000523			BR	16\$	4263
6713	023576	010146		4\$:	MOV	R1,-(SP)	4273
6714	023600	012746	000003		MOV	#3,-(SP)	
6715	023604	010046			MOV	R0,-(SP)	
6716	023606	004767	174254		JSR	PC,RD.DA	
6717	023612	000514			BR	16\$	4263
6718	023614	010146		5\$:	MOV	R1,-(SP)	4276
6719	023616	012746	000012		MOV	#12,-(SP)	
6720	023622	010046			MOV	R0,-(SP)	
6721	023624	004767	174414		JSR	PC,RD.MR	
6722	023630	000505			BR	16\$	4263
6723	023632	010146		6\$:	MOV	R1,-(SP)	4279
6724	023634	012746	000015		MOV	#15,-(SP)	
6725	023640	010046			MOV	R0,-(SP)	
6726	023642	004767	174776		JSR	PC,RD.E1	
6727	023646	000476			BR	16\$	4263
6728	023650	010146		7\$:	MOV	R1,-(SP)	4282
6729	023652	012746	000016		MOV	#16,-(SP)	
6730	023656	010046			MOV	R0,-(SP)	
6731	023660	004767	175200		JSR	PC,RD.E2	
6732	023664	000467			BR	16\$	4263
6733	023666	010146		8\$:	MOV	R1,-(SP)	4285
6734	023670	012746	000010		MOV	#10,-(SP)	
6735	023674	010046			MOV	R0,-(SP)	
6736	023676	004767	174534		JSR	PC,RD.PA	
6737	023702	000460			BR	16\$	4263
6738	023704	010146		9\$:	MOV	R1,-(SP)	4288
6739	023706	012746	000023		MOV	#23,-(SP)	
6740	023712	010046			MOV	R0,-(SP)	
6741	023714	004767	175406		JSR	PC,RD.PD	
6742	023720	000451			BR	16\$	4263
6743	023722	010146		10\$:	MOV	R1,-(SP)	4291
6744	023724	012746	000021		MOV	#21,-(SP)	
6745	023730	010046			MOV	R0,-(SP)	
6746	023732	004767	175572		JSR	PC,RD.EE	
6747	023736	000442			BR	16\$	4263
6748	023740	010146		11\$:	MOV	R1,-(SP)	4294
6749	023742	012746	000022		MOV	#22,-(SP)	
6750	023746	010046			MOV	R0,-(SP)	
6751	023750	004767	175512		JSR	PC,RD.EL	
6752	023754	000433			BR	16\$	4263
6753	023756	010146		12\$:	MOV	R1,-(SP)	4297
6754	023760	012746	000005		MOV	#5,-(SP)	
6755	023764	010046			MOV	R0,-(SP)	

6757								22-Dec-1980 09:24:31	TOPS
6758								22-Dec-1980 09:21:22	PA:<
6759									
6760	023766	004767	176774		JSR	PC, RD.DS			
6761	023772	000424			BR	16\$:		4263
6762	023774	010146		13\$:	MOV	R1, -(SP)	:		4300
6763	023776	012746	000017		MOV	#17, -(SP)	:		
6764	024002	010046			MOV	R0, -(SP)	:		
6765	024004	004767	175644		JSR	PC, RD.D1			
6766	024010	000415			BR	16\$:		4263
6767	024012	010146		14\$:	MOV	R1, -(SP)	:		4303
6768	024014	012746	000020		MOV	#20, -(SP)	:		
6769	024020	010046			MOV	R0, -(SP)	:		
6770	024022	004767	176140		JSR	PC, RD.D2			
6771	024026	000406			BR	16\$:		4263
6772	024030	010146		15\$:	MOV	R1, -(SP)	:		4306
6773	024032	012746	000016		MOV	#16, -(SP)	:		
6774	024036	010046			MOV	R0, -(SP)	:		
6775	024040	004767	176450		JSR	PC, RD.D3			
6776	024044	062706	000006	16\$:	ADD	#6, SP	:		4236
6777	024050	000207			RTS	PC	:		4235
6778									
6779									
6780									
6785									
6786									
6787	:	4310	! <BLF/PAGE></td <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>						

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


```

6789 :ML4
6790 :
6791 :
6792 :          4311 BGNMSG (DUMPER);
6796 :
6800 024052 004767 000004          DUMPER::JSR      PC,MSDUMPER          ;          4311
6801 024056 104423          TRAP      23
6802 024060 000207          RTS       PC
6803 :
6804 :          ; Routine Size: 4 words
6805 :          ; Maximum stack depth per invocation: 0 words
6810 :
6811 :
6812 :          4312
6813 :          4313 !++
6814 :          4314 !
6815 :          4315 !      FUNCTIONAL DESCRIPTION
6816 :          4316 !      UPON COMPLETION OF ERROR MESSAGES
6817 :          4317 !      DUMP OUT ALL PERTINENT DRIVE AND
6818 :          4318 !      RH REGISTERS
6819 :          4319 !
6820 :          4320 PRINTB (ONE_FMT, PHR_14);          !PRINT REGISTER DUMP MESSAGE
6821 :          4321 PRINTB (FMT_23);          !PRINT DUMPER COLUMN HEADINGS
6822 :          4322 PRINTB (FMT_24, REG_1, MLCS1, .MLCS1);          !PRINT OUT THE RH & ML11 REGISTER CONTENTS
6823 :          4323 PRINTB (FMT_24, REG_18, MLWC, .MLWC);
6824 :          4324 PRINTB (FMT_24, REG_19, MLBA, .MLBA);
6825 :          4325 PRINTB (FMT_24, REG_6, MLDA, .MLDA);
6826 :          4326 PRINTB (FMT_24, REG_17, MLCS2, .MLCS2);
6827 :          4327 PRINTB (FMT_24, REG_2, MLDS, .MLDS);
6828 :          4328 PRINTB (FMT_24, REG_3, MLER, .MLER);
6829 :          4329 PRINTB (FMT_24, REG_5, MLAS, .MLAS);
6830 :          4330 PRINTB (FMT_24, REG_4, MLPR, .MLPR);
6831 :          4331 PRINTB (FMT_24, REG_7, MLDT, .MLDT);
6832 :          4332 PRINTB (FMT_24, REG_9, MLSN, .MLSN);
6833 :          4333 PRINTB (FMT_24, REG_14, MLEE, .MLEE);
6834 :          4334 PRINTB (FMT_24, REG_15, MLEL, .MLEL);
6835 :          4335
6836 :          4336 if .LST_DUT_REG eql 21          !SEE IF THIS IS A RH70
6837 :          4337 then          !IF YES THEN PRINT RH70 REGISTERS
6838 :          4338 begin
6839 :          4339 PRINTB (FMT_24, REG_20, MLBAE, .MLBAE);
6840 :          4340 PRINTB (FMT_24, REG_21, MLCS3, .MLCS3);
6841 :          4341 end;
6842 :          4342

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (49)

```

6844 ;ML4
6845 ;
6846 ;
6847 ;      4343  ENDMSG:
6851 ;
6855 024062
6856 024062 012746 007662
6857 024066 012746 006016
6858 024072 012746 000002
6859 024076 010600
6860 024100 104414
6861 024102 012716 005652
6862 024106 012746 000001
6863 024112 010600
6864 024114 104414
6865 024116 017716 167032
6866 024122 016746 167026
6867 024126 012746 010256
6868 024132 012746 005722
6869 024136 012746 000004
6870 024142 010600
6871 024144 104414
6872 024146 017716 167012
6873 024152 016746 167006
6874 024156 012746 010430
6875 024162 012746 005722
6876 024166 012746 000004
6877 024172 010600
6878 024174 104414
6879 024176 017716 166772
6880 024202 016746 166766
6881 024206 012746 010436
6882 024212 012746 005722
6883 024216 012746 000004
6884 024222 010600
6885 024224 104414
6886 024226 017716 166752
6887 024232 016746 166746
6888 024236 012746 010316
6889 024242 012746 005722
6890 024246 012746 000004
6891 024252 010600
6892 024254 104414
6893 024256 017716 166732
6894 024262 016746 166726
6895 024266 012746 010420
6896 024272 012746 005722
6897 024276 012746 000004
    
```

MSDUMPER:

```

MOV #PHR.14,-(SP) ; 4320
MOV #ONE.FMT,-(SP) ;
MOV #2,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV #FMT.23,(SP) ; 4321
MOV #1,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV @ML.REG,(SP) ; 4322
MOV ML.REG,-(SP) ;
MOV #REG.1,-(SP) ;
MOV #FMT.24,-(SP) ;
MOV #4,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV @ML.REG+10,(SP) ; 4323
MOV ML.REG+10,-(SP) ;
MOV #REG.18,-(SP) ;
MOV #FMT.24,-(SP) ;
MOV #4,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV @ML.REG+20,(SP) ; 4324
MOV ML.REG+20,-(SP) ;
MOV #REG.19,-(SP) ;
MOV #FMT.24,-(SP) ;
MOV #4,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV @ML.REG+30,(SP) ; 4325
MOV ML.REG+30,-(SP) ;
MOV #REG.6,-(SP) ;
MOV #FMT.24,-(SP) ;
MOV #4,-(SP) ;
MOV SP,R0 ; SP,*
TRAP 14 ;
MOV @ML.REG+40,(SP) ; 4326
MOV ML.REG+40,-(SP) ;
MOV #REG.17,-(SP) ;
MOV #FMT.24,-(SP) ;
MOV #4,-(SP) ;
    
```

```

6899          :ML4
6900          :
6901
6902 024302 010600      MOV      SP,R0          ; SP,*
6903 024304 104414      TRAP     14
6904 024306 017716 166712  MOV      @ML.REG+50,(SP) ;
6905 024312 016746 166706  MOV      ML.REG+50,-(SP)
6906 024316 012746 010266  MOV      #REG.2,-(SP)
6907 024322 012746 005722  MOV      #FMT.24,-(SP)
6908 024326 012746 000004  MOV      #4,-(SP)
6909 024332 010600      MOV      SP,R0          ; SP,*
6910 024334 104414      TRAP     14
6911 024336 017716 166672  MOV      @ML.REG+60,(SP) ;
6912 024342 016746 166666  MOV      ML.REG+60,-(SP)
6913 024346 012746 010274  MOV      #REG.3,-(SP)
6914 024352 012746 005722  MOV      #FMT.24,-(SP)
6915 024356 012746 000004  MOV      #4,-(SP)
6916 024362 010600      MOV      SP,R0          ; SP,*
6917 024364 104414      TRAP     14
6918 024366 017716 166652  MOV      @ML.REG+70,(SP) ;
6919 024372 016746 166646  MOV      ML.REG+70,-(SP)
6920 024376 012746 010310  MOV      #REG.5,-(SP)
6921 024402 012746 005722  MOV      #FMT.24,-(SP)
6922 024406 012746 000004  MOV      #4,-(SP)
6923 024412 010600      MOV      SP,R0          ; SP,*
6924 024414 104414      TRAP     14
6925 024416 017716 166652  MOV      @ML.REG+120,(SP) ;
6926 024422 016746 166646  MOV      ML.REG+120,-(SP)
6927 024426 012746 010302  MOV      #REG.4,-(SP)
6928 024432 012746 005722  MOV      #FMT.24,-(SP)
6929 024436 012746 000004  MOV      #4,-(SP)
6930 024442 010600      MOV      SP,R0          ; SP,*
6931 024444 104414      TRAP     14
6932 024446 017716 166632  MOV      @ML.REG+130,(SP) ;
6933 024452 016746 166626  MOV      ML.REG+130,-(SP)
6934 024456 012746 010324  MOV      #REG.7,-(SP)
6935 024462 012746 005722  MOV      #FMT.24,-(SP)
6936 024466 012746 000004  MOV      #4,-(SP)
6937 024472 010600      MOV      SP,R0          ; SP,*
6938 024474 104414      TRAP     14
6939 024476 017716 166612  MOV      @ML.REG+140,(SP) ;
6940 024502 016746 166606  MOV      ML.REG+140,-(SP)
6941 024506 012746 010340  MOV      #REG.9,-(SP)
6942 024512 012746 005722  MOV      #FMT.24,-(SP)
6943 024516 012746 000004  MOV      #4,-(SP)
6944 024522 010600      MOV      SP,R0          ; SP,*
6945 024524 104414      TRAP     14
6946 024526 017716 166632  MOV      @ML.REG+210,(SP) ;
6947 024532 016746 166626  MOV      ML.REG+210,-(SP)
6948 024536 012746 010376  MOV      #REG.14,-(SP)
6949 024542 012746 005722  MOV      #FMT.24,-(SP)
6950 024546 012746 000004  MOV      #4,-(SP)
6951 024552 010600      MOV      SP,R0          ; SP,*
6952 024554 104414      TRAP     14
6953 024556 062706 000146  ADD     #146,SP

```

```

6955                                     ;ML4
6956                                     ;
6957                                     ;
6958 024562 017716 166606             MOV    @ML.REG+220,(SP)      ;
6959 024566 016746 166602             MOV    ML.REG+220,-(SP)   ;
6960 024572 012746 010404             MOV    #REG.15,-(SP)    ;
6961 024576 012746 005722             MOV    #FMT.24,-(SP)    ;
6962 024602 012746 000004             MOV    #4,-(SP)         ;
6963 024606 010600                     MOV    SP,R0              ; SP,*
6964 024610 104414                     TRAP   14                  ;
6965 024612 026727 166316 000025      CMP    LST.DUT.REG,#25   ;
6966 024620 001032                     BNE    1$                  ;
6967 024622 017746 166566             MOV    @ML.REG+240,-(SP) ;
6968 024626 016746 166562             MOV    ML.REG+240,-(SP) ;
6969 024632 012746 010444             MOV    #REG.20,-(SP)    ;
6970 024636 012746 005722             MOV    #FMT.24,-(SP)    ;
6971 024640 012746 000004             MOV    #4,-(SP)         ;
6972 024644 010600                     MOV    SP,R0              ; SP,*
6973 024648 104414                     TRAP   14                  ;
6974 024652 017716 166546             MOV    @ML.REG+250,(SP) ;
6975 024656 016746 166542             MOV    ML.REG+250,-(SP) ;
6976 024660 012746 010454             MOV    #REG.21,-(SP)    ;
6977 024664 012746 005722             MOV    #FMT.24,-(SP)    ;
6978 024668 012746 000004             MOV    #4,-(SP)         ;
6979 024672 010600                     MOV    SP,R0              ; SP,*
6980 024700 104414                     TRAP   14                  ;
6981 024702 062706 000022             ADD    #22,SP            ;
6982 024706 062706 000012             ADD    #12,SP            ;
6983 024712 000207                     RTS     PC                  ;
6984
6985                                     ; Routine Size: 205 words
6986                                     ; Maximum stack depth per invocation: 52 words
6991
6992
6993 ;           4344 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

4334

4336

4339

4340

4338

4311

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (50)

```

6995 :ML4
6996 :
6997 :
6998 : 4345 !
6999 : 4346 BGNINIT;
7000 : 4347 !
7001 : 4348 !INITIALIZATION CODE IS EXECUTED AT THE BEGINNING OF EACH
7002 : 4349 !PASS, WHEN POWER DOWN/POWER UP HAS OCCURRED, OR WHEN THE
7003 : 4350 !OPERATOR HAS ISSUED A START, RESTART OR CONTINUE COMMAND.
7004 : 4351 !DURING INITIALIZATION, THE 'GPHARD' MACRO IS USED TO GET
7005 : 4352 !P-TABLE INFORMATION FOR THE LOGICAL UNIT UNDER TEST. THE
7006 : 4353 !NUMBER OF UNITS AVAILABLE FOR TESTING IS CONTAINED IN A
7007 : 4354 !HEADER LOCATION ('LSUNIT').
7008 : 4355 local
7009 : 4356 !OFFSET;
7010 : 4357
7011 : 4358 external
7012 : 4359 !LSUNIT;
7013 : 4360
7014 : 4361 if not READEF (EF_CONTINUE) !SKIP INIT CODE IF CONTINUE
7015 : 4362 then
7016 : 4363 !begin !START GPHARDS AT LUN 0 AND LOAD 'ML_REG'
7017 : 4364
7018 : 4365 !if READEF (EF_START) !SEE IF THIS IS THE VERY FIRST PASS
7019 : 4366 then
7020 : 4367 !begin !THIS IS CATEGORY 1 CODE
7021 : 4368 !ML_LUN = -1;
7022 : 4369
7023 : 4370 !do
7024 : 4371 !begin
7025 : 4372 !ML_LUN = .ML_LUN + 1; !INCREMENT LOGICAL UNIT NUMBER
7026 : 4373
7027 : 4374 !if .ML_LUN geq .LSUNIT then DOCLN; !START OVER IF ALL UNITS HAVE BEEN TESTED
7028 : 4375
7029 : 4376 !end
7030 : 4377 !until (GPHARD (.ML_LUN, PTBL_PTR)) neq 0; !REPEAT THE GPHARD UNTIL A 0 IS RETURNED
7031 : 4378
7032 : 4379 !RH_ADD = ((.PTBL_PTR) + 0); !GET BASE RH ADDRESS FOR THIS UNIT
7033 : 4380 !RH_TYP = ((.PTBL_PTR) + 2); !GET RH TYPE FOR THIS UNIT
7034 : 4381 !RH_VEC = ((.PTBL_PTR) + 4); !GET RH VECTOR FOR THIS UNIT
7035 : 4382 !OFFSET = 0; !INIT OFF SET COUNT
7036 : 4383 !
7037 : 4384 !incr COUNT from 0 to 21 do !LOAD THE REGISTER ADDRESS FOR THIS UNIT INTO ML_REG
7038 : 4385 !begin
7039 : 4386 !ML_REG [.COUNT, REGISTER_ADD] = .RH_ADD + .OFFSET;
7040 : 4387 !OFFSET = .OFFSET + 2;
7041 : 4388 !end;
7042 : 4389
7043 : 4390 !end
7044 : 4391 !else !IS THIS A NEW PASS
7045 : 4392 !begin
7046 : 4393
7047 : 4394 !if READEF (EF_NEW) then ML_LUN = -1; !IF NEW PASS START GPHARDS AT LUN 0
7048 : 4395
7049 : 4396 !do

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (50)

```

7051 :ML4
7052 :
7053 :
7054 :      4397      begin
7055 :      4398      ML_LUN = .ML_LUN + 1;          !IF NOT GET NEXT LUN PTABLE
7056 :      4399
7057 :      4400      if .ML_LUN geq .LSUNIT then DOCLN; !START OVER IF ALL UNITS ARE TESTED
7058 :      4401
7059 :      4402      end
7060 :      4403      until (GP1ARD (.ML_LUN, PTBL_PTR)) neq 0;
7061 :      4404
7062 :      4405      end;
7063 :      4406
7064 :      4407      PAR DIS = .((.PTBL_PTR) + 12);    !GET PARITY DISABLE FLAG
7065 :      4408      ML_DUT = .((.PTBL_PTR) + 10);    !GET DRIVE NUMBER
7066 :      4409      OP_NUM ARR = .((.PTBL_PTR) + 6) - 1; !GET OPERATOR NUMBER OF ARRAYS
7067 :      4410      GOOD_BLK = ZEROES;              !INIT GOOD BLOCK TO BLOCK ZERO
7068 :      4411      ARR_16 = ZEROES;               !INIT ARRAY 16 TO ZERO
7069 :      4412      LST_ARR = ZEROES;              !INIT LAST ARRAY TO ZERO
7070 :      4413      LST_BLK = ZEROES;              !INIT LAST BLOCK TO ZERO
7071 :      4414
7072 :      4415      if .((.PTBL_PTR) + 8) IS_SET    !CALCULATE ML11 16K MOS RAM PARAMETERS
7073 :      4416      then
7074 :      4417      begin
7075 :      4418      DRIVE_TYPE = %o'000110';          !EXPECTED DRIVE TYPE VALUE
7076 :      4419      W_C_SIZE = %o'140000';         !WORD COUNT SIZE FOR 16K WORD XFER
7077 :      4420      RAS_INC = %o'200';            !RAS INCREMENT FOR 16K RAMS
7078 :      4421      CHIP_SIZ = 16;                !CHIP SIZE
7079 :      4422      ARR_INC = %o'1000';           !ARRAY INCREMENT
7080 :      4423      ARR_16<9, 4> = %o'17';       !ARRAY 16
7081 :      4424      LST_ARR<9, 4> = .OP_NUM ARR; !LAST ARRAY
7082 :      4425      LST_BLK<9, 4> = .OP_NUM ARR; !LAST BLOCK
7083 :      4426      LST_BLK = .LST_BLK or %o'777';
7084 :      4427      end
7085 :      4428      else                          !CALCULATE ML11 64K MOS RAM PARAMETERS
7086 :      4429      begin
7087 :      4430      DRIVE_TYPE = %o'000111';          !EXPECTED DRIVE TYPE VALUE
7088 :      4431      W_C_SIZE = %o'000000';         !WORD COUNT SIZE FOR 64K WORD XFER
7089 :      4432      RAS_INC = %o'1000';           !RAS INCREMENT FOR 64K RAMS
7090 :      4433      CHIP_SIZ = 64;                !CHIP SIZE
7091 :      4434      ARR_INC = %o'4000';           !ARRAY INCREMENT
7092 :      4435      ARR_16<11, 4> = %o'74';       !ARRAY 16
7093 :      4436      LST_ARR<11, 4> = .OP_NUM ARR; !LAST ARRAY
7094 :      4437      LST_BLK<11, 4> = .OP_NUM ARR; !LAST BLOCK
7095 :      4438      LST_BLK = .LST_BLK or %o'3777';
7096 :      4439      end;
7097 :      4440
7098 :      4441      if .((.PTBL_PTR) + 2) eql %o'70' then LST_DUT_REG = 21 else LST_DUT_REG = 19;
7099 :      4442
7100 :      4443      PRINTB (FMT_17, .ML_LUN);      !TELL OPERATOR WHICH UNIT IS BEING TESTED
7101 :      4444      CLR_MBUS;                        !CLEAR MASS BUS
7102 :      4445      end;
7103 :      4446
7104 :      4447      ENDINIT;

```

Line	Address	Offset	Value	Label	Instruction	Register/Operand	Comment	Address
7112								
7113								
7114					.GLOBL	LSUNIT		
7115								
7119	024714	004167	157140	LINIT:	JSR	R1,SSAVE3	:	4343
7120	024720	012700	000036		MOV	#36,R0	:	4361
7121	024724	104447			TRAP	47	:	
7122	024726	103001			BHIS	1\$:	
7123	024730	000207			RTS	PC	:	
7124	024732	012700	000040	1\$:	MOV	#40,R0	:	4365
7125	024736	104447			TRAP	47	:	
7126	024740	103061			BHIS	5\$:	
7127	024742	012767	177777	166630	MOV	#-1,ML.LUN	:	4368
7128	024750	005267	166624	2\$:	INC	ML.LUN	:	4372
7129	024754	026767	166620	155030	CMP	ML.LUN,LSUNIT	:	4374
7130	024762	002401			BLT	3\$:	
7131	024764	104444			TRAP	44	:	
7132	024766	016700	166606	3\$:	MOV	ML.LUN,R0	:	4377
7133	024772	104442			TRAP	42	:	
7134	024774	010067	164556		MOV	R0,PTBL.PTR	:	
7135	025000	005767	164552		TST	PTBL.PTR	:	
7136	025004	001761			BEQ	2\$:	
7137	025006	017767	164544	166556	MOV	@PTBL.PTR,RH.ADD	:	4379
7138	025014	016701	164536		MOV	PTBL.PTR,R1	:	4380
7139	025020	016167	000002	166546	MOV	2(R1),RH.TYP	:	
7140	025026	016701	164524		MOV	PTBL.PTR,R1	:	4381
7141	025032	016167	000004	166536	MOV	4(R1),RH.VEC	:	
7142	025040	005002			CLR	R2	: OFFSET	4382
7143	025042	005001			CLR	R1	: COUNT	4384
7144	025044	010100		4\$:	MOV	R1,R0	: COUNT,*	4386
7145	025046	006300			ASL	R0	:	
7146	025050	006300			ASL	R0	:	
7147	025052	006300			ASL	R0	:	
7148	025054	016703	166512		MOV	RH.ADD,R3	:	
7149	025060	060203			ADD	R2,R3	: OFFSET,*	
7150	025062	010360	013154		MOV	R3,ML.REG(R0)	:	
7151	025066	062702	000002		ADD	#2,R2	: *,OFFSET	4387
7152	025072	005201			INC	R1	: COUNT	4384
7153	025074	020127	000025		CMP	R1,#25	: COUNT,*	
7154	025100	003761			BLE	4\$:	
7155	025102	000426			BR	8\$:	4365
7156	025104	012700	000035	5\$:	MOV	#35,R0	:	4394
7157	025110	104447			TRAP	47	:	
7158	025112	103003			BHIS	6\$:	
7159	025114	012767	177777	166456	MOV	#-1,ML.LUN	:	
7160	025122	005267	166452	6\$:	INC	ML.LUN	:	4398

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

7218      :ML4
7219      :
7220
7221 025464 006303      ASL      R3
7222 025466 006303      ASL      R3
7223 025470 006303      ASL      R3
7224 025472 042703 103777 BIC      #103777,R3
7225 025476 042767 074000 164072 BIC      #74000,LST.ARR
7226 025504 050367 164066      BIS      R3,LST.ARR
7227 025510 016703 164044      MOV      OP.NUM.ARR,R3
7228 025514 000303      SWAB    R3
7229 025516 006303      ASL      R3
7230 025520 006303      ASL      R3
7231 025522 006303      ASL      R3
7232 025524 042703 103777 BIC      #103777,R3
7233 025530 042767 074000 164034 BIC      #74000,LST.BLK
7234 025536 050367 164030      BIS      R3,LST.BLK
7235 025542 052767 003777 164022 BIS      #3777,LST.BLK
7236 025550 016701 164002 10$:      MOV      PTBL.PTR,R1
7237 025554 026127 000002 000070      CMP      2(R1),#70
7238 025562 001004      BNE     11$
7239 025564 012767 000025 165342      MOV      #25,LST.DUT.REG
7240 025572 000403      BR      12$
7241 025574 012767 000023 165332 11$:      MOV      #23,LST.DUT.REG
7242 025602 016746 165772 12$:      MOV      ML.LUN,-(SP)
7243 025606 012746 005306      MOV      #FMT.17,-(SP)
7244 025612 012746 000002      MOV      #2,-(SP)
7245 025616 010600      MOV      SP,R0
7246 025620 104414      TRAP    14
7247 025622 152777 000040 165364      BICB    #40,@ML.REG+40
7248 025630 016703 165746      MOV      ML.DUT,R3
7249 025634 042703 177770      BIC      #177770,R3
7250 025640 142777 000007 165346      BICB    #7,@ML.REG+40
7251 025646 150377 165342      BICB    R3,@ML.REG+40
7252 025652 062706 000006      ADD      #6,SP
7253 025656 000207      RTS     PC
7254
7255      : Routine Size: 242 words
7256      : Maximum stack depth per invocation: 7 words
7261
7262
7266
7270 025660 004767 177030      L$INIT::JSR PC,LINIT
7271 025664 104411      TRAP    11

```

4437

4438
4441

4443

4363
4343

4445

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

7273
7274
7275
7276 025666 000207
7277
7278
7279
7284
7285
7286 : 4448 !<BLF/PAGE>

:ML4
:
RTS PC
: Routine Size: 4 words
: Maximum stack depth per invocation: 0 words

1
/

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

7343      ;ML4
7344      ;
7345
7346 025742 001410      1$:      BEQ      4$
7347 025744 016701 154146      MOV      LSDLY,R1      ; *,SSTMP1
7348 025750 001403      BEQ      3$
7349 025752 005016      2$:      CLR      (SP)      ; SSTMP
7350 025754 005301      DEC      R1      ; SSTMP1
7351 025756 001375      BNE      2$
7352 025760 005300      3$:      DEC      R0      ; SSTMP2
7353 025762 000767      BR       1$
7354 025764 032777 177670 165222 4$:      BIT      #-110,2ML.REG+40      ;
7355 025772 001426      BEQ      5$
7356 025774 104455      TRAP     55      ;
7357 025776 000167      .WORD   167
7358 026000 011060      .WORD   RH.ERROR
7359 026002 024052      .WORD   DUMPER
7360 026004 012746 007454      MOV      #PHR.4,-(SP)      ;
7361 026010 012746 010242      MOV      #FNC.23,-(SP)
7362 026014 012746 010420      MOV      #REG.17,-(SP)
7363 026020 012746 006034      MOV      #THR.FMT,-(SP)
7364 026024 012746 000004      MOV      #4,-(SP)
7365 026030 010600      MOV      SP,R0      ; SP,*
7366 026032 104414      TRAP     14
7367 026034 016700 165540      MOV      ML.LUN,R0      ;
7368 026040 104451      TRAP     51
7369 026042 104444      TRAP     44
7370 026044 062706 000012      ADD      #12,SP      ;
7371 026050 005726      5$:      TST      (SP)+      ;
7372 026052 012601      MOV      (SP)+,R1
7373 026054 000207      RTS      PC
7374
7375      ; Routine Size: 59 words
7376      ; Maximum stack depth per invocation: 7 words
7381
7382
7386
7390 026056      T1::
7391 026056 004767 177606      1$:      JSR      PC,$T1      ;
7392 026062 104466      TRAP     66
7393 026064 006000      ROR      R0
7394 026066 103773      BLO      1$
7395 026070 000207      RTS      PC
7396

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

7398 ;ML4
7399 ;
7400 ;
7401 ; Routine Size: 6 words
7402 ; Maximum stack depth per invocation: 0 words
7407 ;
7408 ;
7409 : 4483 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (52)

```

7411 :ML4
7412 :
7413 :
7414 : 4484 !
7415 : 4485 BGNSTST;
7416 : 4486
7417 : 4487 !++
7418 : 4488 TEST NUMBER: TST 2
7419 : 4489
7420 : 4490 TEST NAME: MASS BUS HANDSHAKE TEST
7421 : 4491
7422 : 4492 TEST DESCRIPTION:
7423 : 4493 TEST MASS BUS TO UNIBUS COMMUNICATIONS
7424 : 4494 VIA THE CONTROL BUS BY:
7425 : 4495
7426 : 4496 1. READING RH AND DRIVE REGISTERS AND
7427 : 4497 TEST THE NON EXISTANT DRIVE 'NED'
7428 : 4498 BIT
7429 : 4499
7430 : 4500 !-
7431 : 4501
7432 : 4502 Local
7433 : 4503 SAVE, !TEMPORARY SAVE LOCATION
7434 : 4504 DODU_FLG; !DROP UNIT FLAG
7435 : 4505
7436 : 4506 DODU_FLG = ZERO;
7437 : 4507
7438 : 4508 incr REG_SEL from 0 to .LST_DUT_REG do !TEST ALL PRESENT RH REGISTERS
7439 : 4509 begin
7440 : 4510 BGNSUB; !START OF SCOPE LOOP
7441 : 4511 CLR MBUS;
7442 : 4512 SAVE = ..ML_REG [.REG_SEL, REGISTER_ADD]; !READ THE REGISTER
7443 : 4513
7444 : 4514 if .NED IS_SET !DID READ CAUSE THE NED BIT TO SET
7445 : 4515 then
7446 : 4516 begin !REPORT AN ERROR IF SET
7447 : 4517 ERRDF (120, RH ERROR, DUMPER);
7448 : 4518 PRINTB (FIV_FMT, WRD 62, PHR 5, WRD 12, WRD 52, FNC_6);
7449 : 4519 PRINTB (FMT_11, .ML_REG [.REG_SEL, REGISTER_ADD]);
7450 : 4520 DODU_FLG = ONE;
7451 : 4521 end;
7452 : 4522
7453 : 4523 ENDSUB; !END OF SCOPE LOOP
7454 : 4524 end;
7455 : 4525
7456 : 4526 if .DODU_FLG IS_SET
7457 : 4527 then
7458 : 4528 begin
7459 : 4529 DODU (.ML_LUN);
7460 : 4530 DOCLN;
7461 : 4531 end;
7462 : 4532
7463 : 4533 ENDTST;

```

7471										
7475	026072	004167	156020		\$T2:	JSR	R1,\$SAVE5	:		4482
7476	026076	005001				CLR	R1	:	DODU.FLG	4506
7477	026100	016704	165030			MOV	LST.DUT.REG,R4	:		4508
7478	026104	005002				CLR	R2	:	REG.SEL	
7479	026106	000474				BR	4\$:		
7480	026110	010200			1\$:	MOV	R2,R0	:	REG.SEL,*	4512
7481	026112	006300				ASL	R0	:		
7482	026114	006300				ASL	R0	:		
7483	026116	006300				ASL	R0	:		
7484	026120	010003				MOV	R0,R3	:		
7485	026122	104402			2\$:	TRAP	2	:		4509
7486	026124	152777	000040	165062		BISB	#40,@ML.REG+40	:		4510
7487	026132	016700	165444			MOV	ML.DUT,R0	:		
7488	026136	042700	177770			BIC	#177770,R0	:		
7489	026142	142777	000007	165044		BICB	#7,@ML.REG+40	:		
7490	026150	150077	165040			BISB	R0,@ML.REG+40	:		
7491	026154	017305	013154			MOV	@ML.REG(R3),R5	:	*,SAVE	4512
7492	026160	032777	010000	165026		BIT	#10000,@ML.REG+40	:		4514
7493	026166	001440				BEQ	3\$:		
7494	026170	104455				TRAP	55	:		4517
7495	026172	000170				.WORD	170	:		
7496	026174	011060				.WORD	RH.ERROR	:		
7497	026176	024052				.WORD	DUMPER	:		
7498	026200	012746	010004			MOV	#FNC.6,-(SP)	:		4518
7499	026204	012746	007066			MOV	#WRD.52,-(SP)	:		
7500	026210	012746	006426			MOV	#WRD.12,-(SP)	:		
7501	026214	012746	007472			MOV	#PHR.5,-(SP)	:		
7502	026220	012746	007210			MOV	#WRD.62,-(SP)	:		
7503	026224	012746	006062			MOV	#FIV.FMT,-(SP)	:		
7504	026230	012746	000006			MOV	#6,-(SP)	:		
7505	026234	010600				MOV	SP,R0	:	SP,*	
7506	026236	104414				TRAP	14	:		
7507	026240	016316	013154			MOV	ML.REG(R3),(SP)	:		4519
7508	026244	012746	004750			MOV	#FMT.11,-(SP)	:		
7509	026250	012746	000002			MOV	#2,-(SP)	:		
7510	026254	010600				MOV	SP,R0	:	SP,*	
7511	026256	104414				TRAP	14	:		
7512	026260	012701	000001			MOV	#1,R1	:	*,DODU.FLG	4520
7513	026264	062706	000022			ADD	#22,SP	:		4516
7514	026270	104467			3\$:	TRAP	67	:		4521
7515	026272	006000				ROR	R0	:		
7516	026274	103712				BLO	2\$:		
7517	026276	005202				INC	R2	:	REG.SEL	4508
7518	026300	020204			4\$:	CMP	R2,R4	:	REG.SEL,*	
7519	026302	003702				BLE	1\$:		
7520	02 304	005301				DEC	R1	:	DODU.FLG	4526

```
7522      ;ML4
7523      ;
7524
7525 026306 001004      BNE      5$
7526 026310 016700 165264  MOV     ML,LUN,RO      ;
7527 026314 104451      TRAP    51
7528 026316 104444      TRAP    44
7529 026320 000207 5$:      RTS     PC      ;
7530
7531      ; Routine Size: 76 words
7532      ; Maximum stack depth per invocation: 15 words
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544 026322      T2::
7545 026322 004767 177544 1$:      JSR     PC,$T2      ;
7546 026326 104466      TRAP    66
7547 026330 006000      ROR     R0
7548 026332 103773      BLO    1$
7549 026334 000207      RTS     PC
7550
7551
7552      ; Routine Size: 6 words
7553      ; Maximum stack depth per invocation: 0 words
7554
7555
7556
7557
7558
7559
7560
7561 ;      4534 !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

4529

4482

4531

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (53)

```
7563 :ML4
7564 :
7565 :
7566 : 4535
7567 : 4536 BGNTST;
7568 : 4537
7569 : 4538 !++
7570 : 4539 TEST NUMBER: TST 3
7571 : 4540
7572 : 4541 TEST NAME: DRIVE PRESENT TEST
7573 : 4542
7574 : 4543 TEST DESCRIPTION:
7575 : 4544 THIS TEST READS THE DESIRED SECTOR
7576 : 4545 ADDRESS REGISTER OF THE DRIVE UNDER
7577 : 4546 TEST, DELAYS 100 US, THEN
7578 : 4547 READS THE NED BIT OF MLCS2
7579 : 4548
7580 : 4549 IF SET, AN ERROR MESSAGE IS
7581 : 4550 PRINTED AND THE UNIT IS DROPPED
7582 : 4551
7583 : 4552 !--
7584 : 4553
7585 : 4554 local
7586 : 4555 DODU_FLG, !DROP UNIT FLAG
7587 : 4556 SAVE; !TEMP STORAGE LOCATION
7588 : 4557
7589 : 4558 DODU_FLG = ZERO;
7590 : 4559 BGNSUB;
7591 : 4560 CLR MBUS;
7592 : 4561 SAVE = .MLDA; !READ A DRIVE REGISTER
7593 : 4562 DELAY (ONE_US); !DELAY 1 US
7594 : 4563
7595 : 4564 if .NED IS_SET !TEST THE NED BIT
7596 : 4565 then
7597 : 4566 begin
7598 : 4567 ERRDF (1, ASYNC, DUMPER); !IF SET THEN REPORT ERROR AND SET DODU_FLG
7599 : 4568 PRINTB (ONE_FMT, PHR_3);
7600 : 4569 DODU_FLG = ONE;
7601 : 4570 end;
7602 : 4571
7603 : 4572 ENDSUB;
7604 : 4573
7605 : 4574 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU IS_SET
7606 : 4575 then
7607 : 4576 begin
7608 : 4577 DODU (.ML_LUN);
7609 : 4578 DOCLN;
7610 : 4579 end;
7611 : 4580
7612 : 4581 ENDTST;
7616 :
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

```

7618      ;ML4
7619      ;
7620      ;
7624 026336 004167 155534      $T3:   JSR     R1,$SAVE4      ;
7625 026342 005746              TST     -(SP)          ;
7626 026344 005003              CLR     R3             ; DODU.FLG
7627 026346 104402              TRAP   2              ;
7628 026350 152777 000040 164636  BISB   #40,@ML.REG+40  ;
7629 026356 016702 165220      MOV     ML,DUT,R2      ;
7630 026362 042702 177770      BIC    #177770,R2     ;
7631 026366 142777 000007 164620  BICB   #7,@ML.REG+40  ;
7632 026374 150277 164614      BISB   R2,@ML.REG+40  ;
7633 026400 017704 164600      MOV     @ML.REG+30,R4 ; *,SAVE
7634 026404 012701 000001      MOV     #1,R1         ; *,SSTMP2
7635 026410 001410              BEQ    5$             ;
7636 026412 016702 153500      MOV     LSDLY,R2     ; *,SSTMP1
7637 026416 001403              BEQ    4$             ;
7638 026420 005016              CLR    (SP)          ; SSTMP
7639 026422 005302              DEC    R2             ; SSTMP1
7640 026424 001375              BNE    3$             ;
7641 026426 005301              DEC    R1             ; SSTMP2
7642 026430 000767              BR     2$             ;
7643 026432 032777 010000 164554 5$:   BIT    #10000,@ML.REG+40 ;
7644 026440 001420              BEQ    6$             ;
7645 026442 104455              TRAP   55            ;
7646 026444 000001              .WORD  1              ;
7647 026446 010464              .WORD  ASYNC          ;
7648 026450 024052              .WORD  DUMPER        ;
7649 026452 012746 007422      MOV     #PHR.3,-(SP)  ;
7650 026456 012746 006016      MOV     #ONE.FMT,-(SP) ;
7651 026462 012746 000002      MOV     #2,-(SP)     ;
7652 026466 010600              MOV     SP,R0        ; SP,*
7653 026470 104414              TRAP   14            ;
7654 026472 012703 000001      MOV     #1,R3         ; *,DODU.FLG
7655 026476 062706 000006      ADD    #6,SP         ;
7656 026502 104467              TRAP   67            ;
7657 026504 006000              ROR    R0             ;
7658 026506 103717              BLO    1$            ;
7659 026510 005303              DEC    R3             ; DODU.FLG
7660 026512 001004              BNE    7$            ;
7661 026514 016700 165060      MOV     ML,LUN,KO    ;
7662 026520 104451              TRAP   51            ;
7663 026522 104444              TRAP   44            ;
7664 026524 005726              TST    (SP)+         ;
7665 026526 000207              RTS     PC            ;

```

7666
7667
7668

; Routine Size: 61 words
; Maximum stack depth per invocation: 9 words

7674 ;ML4

7675 ;

7676 ;

7680

7684 026530

7685 026530 004767 177602

7686 026534 104466

7687 026536 006000

7688 026540 103773

7689 026542 000207

7690

7691

7692

7697

7698

7699 ; 4582 !<BLF/PAGE>

22-Dec-1980 09:24:31

22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)

PA:<NEALE>BL3ML4.BLI.2 (53)

T3::

1\$: JSR PC,\$T3 ;

TRAP 66

ROR R0

BLO 1\$

RTS PC

4579

; Routine Size: 6 words

; Maximum stack depth per invocation: 0 words

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (54)

```

7701 :ML4
7702 :
7703 :
7704 : 4583 !
7705 : 4584 BGNTST;
7706 : 4585
7707 : 4586 !++
7708 : 4587 TEST NUMBER: TST 4
7709 : 4588
7710 : 4589 TEST NAME: DRIVE SELECTION TEST
7711 : 4590
7712 : 4591 TEST DESCRIPTION:
7713 : 4592 THIS TEST TESTS FOR UNIQUE DRIVE SELECTION BY WRITING THE DRIVE
7714 : 4593 UNDER TEST (DUT) DRIVE NUMBER INTO ITS DSA REG. THEN WRITING THE DRIVE
7715 : 4594 NUMBERS OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REGISTERS
7716 : 4595 AND READING THE DUT DSA FOR ITS DRIVE NUMBER.
7717 : 4596
7718 : 4597 IMPLICIT INPUTS:
7719 : 4598 ML_DUT
7720 : 4599 LOADED DURING THE INITIALIZATION CODE AND CONTAINS THE DRIVE
7721 : 4600 NUMBER OF THE DRIVE PRESENTLY BEING TESTED.
7722 : 4601 !--
7723 : 4602
7724 : 4603 local
7725 : 4604 DODU_FLG, !DROP UNIT FLAG
7726 : 4605 SAVE; !TEMPORARY SAVE LOCATION
7727 : 4606
7728 : 4607 BGNSUB;
7729 : 4608 CLR MBUS;
7730 : 4609 DODU_FLG = ZERO;
7731 : 4610 MLDA = .ML_DUT; !LOAD THIS DRIVES DRIVE NO. INTO ITS DSA REG
7732 : 4611 SAVE = .MLDA; !READ THE REGISTER BACK
7733 : 4612
7734 : 4613 if .SAVE neq .ML_DUT !SEE IF DSA HAS DRIVE NUMBER
7735 : 4614 then
7736 : 4615 begin
7737 : 4616 ERRDF (2, INTER, DUMPER); !ERROR AND EXIT TEST IF DSA NEQ DRIVE NUM
7738 : 4617 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 3, WRD 37, WRD 13, REG_6);
7739 : 4618 PRINTB (FMT_2, .ML_DOT, .SAVE, (.ML_DUT xof .SAVE));
7740 : 4619 EXIT_TST;
7741 : 4620 end;
7742 : 4621
7743 : 4622 incr DRV_SEL from 0 to 7 do !WRITE DRV NO OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REG.
7744 : 4623
7745 : 4624 if .DRV_SEL neq .ML_DUT !SKIP IF .DRV_SEL EQL TO THE DRIVE UNDER TEST (DUT)
7746 : 4625 then
7747 : 4626 begin
7748 : 4627 DRV_NUM = .DRV_SEL; !SELECT DRIVE TO BE WRITTEN TO
7749 : 4628 MLDA = .DRV_SEL; !WRITE DRIVE SEL NO. INTO ITS DSA REG
7750 : 4629 DELAY (ONE_OS); !DELAY 1 US
7751 : 4630 end;
7752 : 4631
7753 : 4632 DRV_NUM = .ML_DUT; !SELECT THE DUT
7754 : 4633 SAVE = .MLDA; !READ ITS DSA REG
7755 : 4634

```

```

7757 :ML4
7758 :
7759 :
7760 :      4635 if .SAVE neq .ML_DUT
7761 :      4636 then
7762 :      4637   begin
7763 :      4638     ERRDF (3, ASYNC, DUMPER);
7764 :      4639     PRINTB (THR_FMT, FNC 3, WRD 37, WRD_14);
7765 :      4640     PRINTB (FMT_1, .ML_DOT, .SAVE);
7766 :      4641     DODU_FLG = ONE;
7767 :      4642     end;
7768 :      4643
7769 :      4644   ENDSUB;
7770 :      4645
7771 :      4646 if .DODU_FLG IS_SET
7772 :      4647 then
7773 :      4648   begin
7774 :      4649     DODU (.ML_LUN);
7775 :      4650     DOCLN;
7776 :      4651     end;
7777 :      4652
7778 :      4653   ENDTST;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (54)

!SEE IF WRITTING 10 OTHER DRIVES CHANGE ITS VALUE

!ERROR AND SET DODU_FLG IF CHANGED

!DROP UNIT IF DODU_FLG IS_SET

7786	026544	004167	155546	\$T4:	JSR	R1,\$SAVE5	:	4581
7787	026550	005746			TST	-(SP)	:	
7788	026552	104402		1\$:	TRAP	2	:	4605
7789	026554	152777	000040	164432	BISB	#40,@ML.REG+40	:	4607
7790	026562	016703	165014		MOV	ML.DUT,R3	:	
7791	026566	042703	177770		BIC	#177770,R3	:	
7792	026572	142777	000007	164414	BICB	#7,@ML.REG+40	:	
7793	026600	150377	164410		BISB	R3,@ML.REG+40	:	
7794	026604	005005			CLR	R5	: DODU.FLG	4609
7795	026606	016777	164770	164370	MOV	ML.DUT,@ML.REG+30	:	4610
7796	026614	017704	164364		MOV	@ML.REG+30,R4	: *,SAVE	4611
7797	026620	020467	164756		COMP	R4,ML.DUT	: SAVE,*	4613
7798	026624	001451			BEQ	2\$:	
7799	026626	104455			TRAP	55	:	4616
7800	026630	000002			.WORD	2	:	
7801	026632	010672			.WORD	INTER	:	
7802	026634	024052			.WORD	DUMPER	:	
7803	026636	012746	010316		MOV	#REG.6,-(SP)	:	4617
7804	026642	012746	006436		MOV	#WRD.13,-(SP)	:	
7805	026646	012746	006700		MOV	#WRD.37,-(SP)	:	
7806	026652	012746	007750		MOV	#FNC.3,-(SP)	:	
7807	026656	012746	006426		MOV	#WRD.12,-(SP)	:	
7808	026662	012746	007454		MOV	#PHR.4,-(SP)	:	
7809	026666	012746	006100		MOV	#SIX.FMT,-(SP)	:	
7810	026672	012746	000007		MOV	#7,-(SP)	:	

Address	OpCode	Operand 1	Operand 2	Comment	Label
7812					
7813					
7814					
7815	026676	010600		MOV SP,R0	: SP,*
7816	026700	104414		TRAP 14	
7817	026702	016716	164674	MOV ML.DUT,(SP)	
7818	026706	010403		MOV R4,R3	: SAVE,*
7819	026710	041603		BIC (SP),R3	
7820	026712	040416		BIC R4,(SP)	: SAVE,*
7821	026714	050316		BIS R3,(SP)	
7822	026716	010446		MOV R4,-(SP)	: SAVE,*
7823	026720	016746	164656	MOV ML.DUT,-(SP)	
7824	026724	012746	004266	MOV #FMT.2,-(SP)	
7825	026730	012746	000004	MOV #4,-(SP)	
7826	026734	010600		MOV SP,R0	: SP,*
7827	026736	104414		TRAP 14	
7828	026740	104463		TRAP 63	
7829	026742	062706	000030	ADD #30,SP	
7830	026746	000523		BR 10\$	
7831	026750	005003		2\$: CLR R3	: DRV.SEL
7832	026752	020367	164624	3\$: CMP R3,ML.DUT	: DRV.SEL,*
7833	026756	001425		BEQ 7\$	
7834	026760	010302		MOV R3,R2	: DRV.SEL,*
7835	026762	042702	177770	BIC #177770,R2	
7836	026766	142777	000007	164220 BICB #7,@ML.REG+40	
7837	026774	150277	164214	BISB R2,@ML.REG+40	
7838	027000	010377	164200	MOV R3,@ML.REG+30	: DRV.SEL,*
7839	027004	012701	000001	MOV #1,R1	: *,SSTMP2
7840	027010	001410		4\$: BEQ 7\$	
7841	027012	016702	153100	MOV LSDLY,R2	: *,SSTMP1
7842	027016	001403		BEQ 6\$	
7843	027020	005016		5\$: CLR (SP)	: SSTMP
7844	027022	005302		DEC R2	: SSTMP1
7845	027024	001375		BNE 5\$	
7846	027026	005301		6\$: DEC R1	: SSTMP2
7847	027030	000767		BR 4\$	
7848	027032	005203		7\$: INC R3	: DRV.SEL
7849	027034	020327	000007	CMP R3,#7	: DRV.SEL,*
7850	027040	003744		BLE 3\$	
7851	027042	016703	164534	MOV ML.DUT,R3	
7852	027046	042703	177770	BIC #177770,R3	
7853	027052	142777	000007	164134 BICB #7,@ML.REG+40	
7854	027060	150377	164130	BISB R3,@ML.REG+40	
7855	027064	017704	164114	MOV @ML.REG+30,R4	: *,SAVE
7856	027070	020467	164506	CMP R4,ML.DUT	: SAVE,*
7857	027074	001435		BEQ 8\$	
7858	027076	104455		TRAP 55	
7859	027100	000003		.WORD 3	
7860	027102	010464		.WORD ASYNC	
7861	027104	024052		.WORD DUMPER	
7862	027106	012746	006442	MOV #WORD.14,-(SP)	
7863	027112	012746	006700	MOV #WORD.37,-(SP)	
7864	027116	012746	007750	MOV #FNC.3,-(SP)	
7865	027122	012746	006034	MOV #THR.FMT,-(SP)	
7866	027126	012746	000004	MOV #4,-(SP)	

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

7868      ;ML4
7869      :
7870
7871 027132 010600      MOV      SP,R0      ; SP,*
7872 027134 104414      TRAP     14
7873 027136 010416      MOV      R4,(SP)    ; SAVE,*      4640
7874 027140 016746 164436  MOV      ML.DUT,-(SP)
7875 027144 012746 004222  MOV      #FMT.1,-(SP)
7876 027150 012746 000003  MOV      #3,-(SP)
7877 027154 010600      MOV      SP,R0      ; SP,*
7878 027156 104414      TRAP     14
7879 027160 012705 000001  MOV      #1,R5      ; *,DODU.FLG  4641
7880 027164 062706 000020  ADD      #2),SP      ;              4637
7881 027170 104467      8$: TRAP     67      ;              4642
7882 027172 006000      ROR      R0
7883 027174 103002      BHIS     9$
7884 027176 000167 177350  JMP      1$
7885 027202 005305      9$: DEC      R5      ; DODU.FLG  4646
7886 027204 001004      BNE     10$
7887 027206 016700 164366  MOV      ML.LUN,R0  ;              4649
7888 027212 104451      TRAP     51
7889 027214 104444      TRAP     44
7890 027216 005726      10$: TST      (SP)+   ;              4581
7891 027220 000207      RTS      PC
7892
7893      ; Routine Size: 151 words
7894      ; Maximum stack depth per invocation: 19 words
7895
7896
7897
7898
7899
7900
7901
7902
7903
7904
7905
7906
7907
7908 027222      T4::
7909 027222 004767 177316      1$: JSR      PC,$T4   ;              4651
7910 027226 104466      TRAP     66
7911 027230 006000      ROR      R0
7912 027232 103773      BLO     1$
7913 027234 000207      RTS      PC
7914
7915      ; Routine Size: 6 words
7916      ; Maximum stack depth per invocation: 0 words
7917
7918
7919
7920
7921 :          4654 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (55)

```

7923 :ML4
7924 :
7925 :
7926 : 4655 !
7927 : 4656 BGNTST;
7928 : 4657
7929 : 4658 !++
7930 : 4659 TEST NUMBER: TST 5
7931 : 4660
7932 : 4661 TEST NAME: READ WRITE REG ONES/ZEROES TEST
7933 : 4662
7934 : 4663 TEST DESCRIPTION:
7935 : 4664 THIS TEST WRITES AND READS A DATA PATTERN OF ALL ONES AND ZEROES TO ALL
7936 : 4665 OF THE ML11'S READ / WRITE REGISTERS.
7937 : 4666
7938 : 4667 ROUTINES WRT_REG AND RD_REG ACCEPT ARGUMENTS TO FURTHER SELECT ROUTINES
7939 : 4668 WHICH ACTUALLY PERFORMS THE READING AND WRITING OF THE REGISTERS.
7940 : 4669
7941 : 4670 THE UNIT IS DROPPED ON DETECTED ERRORS.
7942 : 4671
7943 : 4672 IMPLICIT INPUTS:
7944 : 4673 WT DATA
7945 : 4674 LOADED BY READ REGISTER ROUTINES AND CONTAINS THE DATA PATTERN WRITTEN
7946 : 4675 TO THE REGISTERS (REPRESENTS GOOD DATA).
7947 : 4676
7948 : 4677 RD DATA
7949 : 4678 LOADED BY THE READ REGISTER ROUTINES AND CONTAINS THE DATA PATTERN
7950 : 4679 READ FROM THE REGISTER (REPRESENTS BAD DATA).
7951 : 4680
7952 : 4681
7953 : 4682 local
7954 : 4683 ERR_FLG, !ERROR FLAG PASSED TO ROUTINES
7955 : 4684 TST_PAT, !TEST PATTERN
7956 : 4685 index, !POINTS TO REGISTER PRESENTLY BEING TESTED
7957 : 4686 DODU_FLG; !DROP UNIT FLAG
7958 : 4687
7959 : 4688 DODU_FLG = ZERO;
7960 : 4689 TST_PAT = ONES; !LOAD TEST PAT WITH ONES
7961 : 4690
7962 : 4691 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
7963 : 4692 begin
7964 : 4693
7965 : 4694 incr REG_SEL from 0 to 7 do !TEST ELEVEN WRITE/READ REGISTERS
7966 : 4695 begin
7967 : 4696 BGNSUB;
7968 : 4697 CLR_MBUS;
7969 : 4698 WRT_REG (.TST_PAT, .REG_SEL, index); !WRITE TO THE REGISTER
7970 : 4699 RD_REG (.TST_PAT, .REG_SEL, ERR_FLG); !READ THE REGISTER
7971 : 4700
7972 : 4701 if .ERR_FLG IS_SET !SEE IF READ FOUND AN ERROR
7973 : 4702 then
7974 : 4703 begin !IF ERROR FLAG IS_SET THEN ERROR AND SET DODU_FLG
7975 : 4704
7976 : 4705 selectone .REG_SEL of
7977 : 4706 set

```


22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (55)

```

7979 :ML4
7980 :
7981 :
7982 : 4707
7983 : 4708 [0 to 3] :
7984 : 4709 ERRDF (4, ASYNC, DUMPER); !ASYNC MODULE FAILURE
7985 : 4710
7986 : 4711 [4 to 5] :
7987 : 4712 ERRDF (4, SYNC, DUMPER); !SYNC MODULE FAILURE
7988 : 4713
7989 : 4714 [6 to 7] :
7990 : 4715 ERRDF (4, ARR_DAT, DUMPER); !ARRAY DATA MODULE FAILURE
7991 : 4716 tes:
7992 : 4717
7993 : 4718 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
7994 : 4719 PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .QT_DATA, .RD_DATA);
7995 : 4720 DODU_FLG = ONE;
7996 : 4721 end;
7997 : 4722
7998 : 4723 ENDSUB;
7999 : 4724 end;
8000 : 4725
8001 : 4726 TST PAT = not .TST_PAT; !REPEAT AGAIN WITH COMPLIMENT DATA
8002 : 4727 end;
8003 : 4728
8004 : 4729 if .DODU_FLG IS_SET !DROP THIS UNIT IF THE DODU_FLG IS_SET
8005 : 4730 then
8006 : 4731 begin
8007 : 4732 DODU (.ML_LUN);
8008 : 4733 DOCLN;
8009 : 4734 end;
8010 : 4735
8011 : 4736 ENDTST;
8015 :
8019 027236 004167 154634 $T5: JSR R1,SSAVE4 ; 4653
8020 027242 024646 CMP -(SP),-(SP) ;
8021 027244 005004 CLR R4 ; DODU.FLG 4688
8022 027246 012702 177777 MOV #-1,R2 ; *.TST.PAT 4689
8023 027252 005001 CLR R1 ; TWICE 4691
8024 027254 005003 1$: CLR R3 ; REG.SEL 4694
8025 027256 104402 2$: TRAP 2 ; 4695
8026 027260 152777 000040 163726 BISB #40,@ML.REG+40 ; 4696
8027 027266 016700 164310 MOV ML.DUT,R0
8028 027272 042700 177770 BIC #177770,R0
8029 027276 142777 000007 163710 BICB #7,@ML.REG+40
8030 027304 150077 163704 BISB R0,@ML.REG+40
8031 027310 010246 MOV R2,-(SP) ; TST.PAT,* 4698
8032 027312 010346 MOV R3,-(SP) ; REG.SEL,*

```

```

8034      ;ML4
8035      ;
8036
8037 027314 012746 000010      MOV      #10,-(SP)
8038 027320 060616      ADD      SP,(SP)      ; INDEX,*
8039 027322 004767 173546      JSR      PC,WRT.REG
8040 027326 010216      MOV      R2,(SP)      ; TST.PAT,*
8041 027330 010346      MOV      R3,-(SP)      ; REG.SEL,*
8042 027332 012746 000012      MOV      #12,-(SP)
8043 027336 060616      ADD      SP,(SP)      ; ERR.FLG,*
8044 027340 004767 174116      JSR      PC,RD.REG
8045 027344 026627 000012 000001      CMP      12(SP),#1      ; ERR.FLG,*
8046 027352 001106      BNE      6$
8047 027354 005703      TST      R3      ; REG.SEL
8048 027356 002410      BLT      3$
8049 027360 020327 000003      CMP      R3,#3      ; REG.SEL,*
8050 027364 003005      BGT      3$
8051 027366 104455      TRAP     55
8052 027370 000004      .WORD   4
8053 027372 010464      .WORD   ASYNC
8054 027374 024052      .WORD   DUMPER
8055 027376 000425      BR
8056 027400 020327 000004      3$:    CMP      R3,#4      ; REG.SEL,*
8057 027404 002410      BLT      4$
8058 027406 020327 000005      CMP      R3,#5      ; REG.SEL,*
8059 027412 003005      BGT      4$
8060 027414 104455      TRAP     55
8061 027416 000004      .WORD   4
8062 027420 010526      .WORD   SYNC
8063 027422 024052      .WORD   DUMPER
8064 027424 000412      BR
8065 027426 020327 000006      4$:    CMP      R3,#6      ; REG.SEL,*
8066 027432 002407      BLT      5$
8067 027434 020327 000007      CMP      R3,#7      ; REG.SEL,*
8068 027440 003004      BGT      5$
8069 027442 104455      TRAP     55
8070 027444 000004      .WORD   4
8071 027446 010570      .WORD   ARR.DAT
8072 027450 024052      .WORD   DUMPER
8073 027452 012746 007122      5$:    MOV      #WORD.56,-(SP)
8074 027456 012746 007066      MOV      #WORD.52,-(SP)
8075 027462 012746 010004      MOV      #FNC.6,-(SP)
8076 027466 012746 007774      MOV      #FNC.5,-(SP)
8077 027472 12746 006426      MOV      #WORD.12,-(SP)
8078 027476 012746 007454      MOV      #PHR.4,-(SP)
8079 027502 012746 006100      MOV      #SIX.FMT,-(SP)
8080 027506 012746 000007      MOV      #7,-(SP)
8081 027512 010600      MOV      SP,R0      ; SP,*
8082 027514 104414      TRAP     14
8083 027516 016716 163406      MOV      RD.DATA,(SP)
8084 027522 016746 163400      MOV      WT.DATA,-(SP)
8085 027526 016600 000036      MOV      36(SP),R0      ; INDEX,*
8086 027532 006300      ASL      R0
8087 027534 006300      ASL      R0
8088 027536 006300      ASL      R0

```

```

8090      ;ML4
8091      ;
8092
8093 027540 016046 013154      MOV      ML.REG(R0),-(SP)
8094 027544 012746 005216      MOV      #FMT.16, -(SP)
8095 027550 012746 000004      MOV      #4, -(SP)
8096 027554 010600              MOV      SP,R0                ; SP,*
8097 027556 104414              TRAP     14
8098 027560 012704 000001      MOV      #1,R4                ; *,DODU.FLG
8099 027564 062706 000030      ADD      #30,SP                ;
8100 027570 062706 000012      ADD      #12,SP                ;
8101 027574 104467              TRAP     67                    ;
8102 027576 006000              ROR      R0
8103 027600 103626              BLO     2$
8104 027602 005203              INC      R3                    ; REG.SEL
8105 027604 020327 000007      CMP      R3,#7                 ; REG.SEL,*
8106 027610 003622              BLE     2$
8107 027612 005102              COM     R2                    ; TST.PAT
8108 027614 005201              INC      R1                    ; TWICE
8109 027616 020127 000001      CMP      R1,#1                 ; TWICE,*
8110 027622 003614              BLE     1$
8111 027624 005304              DEC     R4                    ; DODU.FLG
8112 027626 001004              BNE     7$
8113 027630 016700 163744      MOV      ML.LUN,R0
8114 027634 104451              TRAP     51
8115 027636 104444              TRAP     44
8116 027640 022626 7$:      CMP      (SP)+,(SP)+
8117 027642 000207              RTS     PC
8118
8119      ; Routine Size: 131 words
8120      ; Maximum stack depth per invocation: 24 words
8125
8126
8130
8134 027644      T5::
8135 027644 004767 177366      1$:      JSR      PC,$T5
8136 027650 104466              TRAP     66
8137 027652 006000              ROR      R0
8138 027654 103773              BLO     1$
8139 027656 000207              RTS     PC
8140
8141      ; Routine Size: 6 words
8142      ; Maximum stack depth per invocation: 0 words

```

8151
8152
8153 ; 4737 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (56)

```
8155 :ML4
8156 :
8157 :
8158 : 4738 !
8159 : 4739 BGNTST;
8160 : 4740
8161 : 4741 !++
8162 : 4742 TEST NUMBER: TST 6
8163 : 4743
8164 : 4744 TEST NAME: READ WRITE REG SHIFTING ONES/ZEROES TEST
8165 : 4745
8166 : 4746 TEST DESCRIPTION:
8167 : 4747 THIS TEST WRITES AND READS A
8168 : 4748 SHIFTING ONE'S AND SHIFTING ZEROE'S
8169 : 4749 PATTERN TO ALL THE ML11'S
8170 : 4750 READ/WRITE REGISTERS
8171 : 4751
8172 : 4752 ROUTINES WRT REG AND RD REG
8173 : 4753 ACCEPT ARGUMENTS TO FURTHER
8174 : 4754 SELECT ROUTINES WHICH ACTUALLY
8175 : 4755 PERFORMS THE READING AND
8176 : 4756 WRITING OF THE REGISTERS.
8177 : 4757
8178 : 4758 THE DRIVE IS DROPPED ON DETECTED
8179 : 4759 ERRORS.
8180 : 4760
8181 : 4761 IMPLICIT INPUTS:
8182 : 4762 WT DATA
8183 : 4763 [LOADED BY READ REGISTER ROUTINES AND
8184 : 4764 CONTAINS THE DATA PATTERN WRITTEN TO THE
8185 : 4765 REGISTERS (REPRESENTS GOOD DATA).
8186 : 4766
8187 : 4767 RD DATA
8188 : 4768 [LOADED BY THE READ REGISTER ROUTINES AND
8189 : 4769 CONTAINS THE DATA PATTERN READ FROM THE
8190 : 4770 REGISTER (REPRESENTS BAD DATA).
8191 : 4771
8192 : 4772 !-
8193 : 4773
8194 : 4774
8195 : 4775 local
8196 : 4776 ERR_FLG,
8197 : 4777 TST_PAT,
8198 : 4778 index,
8199 : 4779 DODU_FLG;
8200 : 4780
8201 : 4781 DODU_FLG = ZERO;
8202 : 4782 TST_PAT = ONE;
8203 : 4783
8204 : 4784 incr SHIFT from 0 to 15 do
8205 : 4785 begin
8206 : 4786
8207 : 4787 incr TWICE from 0 to 1 do
8208 : 4788 begin
8209 : 4789
```

!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

```

8211 :ML4
8212 :
8213 :
8214 : 4790      incr REG_SEL from 0 to 7 do          !TEST ELEVEN READ/WRITE REGISTERS
8215 : 4791      begin
8216 : 4792      BGN SUB;
8217 : 4793      CLR_MBUS;
8218 : 4794      WRT_REG (.TST_PAT, .REG_SEL, index);      !WRITE TO THE REGISTER
8219 : 4795      RD_REG (.TST_PAT, .REG_SEL, ERR_FLG);      !READ THE REGISTER
8220 : 4796
8221 : 4797      if .ERR_FLG IS_SET          !SEE IF THE READ FOUND AN ERROR
8222 : 4798      then
8223 : 4799      begin          !IF THE ERROR FLAG IS_SET THEN ERROR
8224 : 4800
8225 : 4801      select one .REG_SEL of          !FIND WHICH MODULE FAILED
8226 : 4802      set
8227 : 4803
8228 : 4804      [0 to 3] :
8229 : 4805      ERRDF (5, ASYNC, DUMPER);          !ASYNC MODULE FAILURE
8230 : 4806
8231 : 4807      [4 to 5] :
8232 : 4808      ERRDF (5, SYNC, DUMPER);          !SYNC MODULE FAILURE
8233 : 4809
8234 : 4810      [6 to 7] :
8235 : 4811      ERRDF (5, ARR_DAT, DUMPER);      !ARRAY DATA MODULE FAILURE
8236 : 4812
8237 : 4813      tes;
8238 : 4814      PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
8239 : 4815      PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
8240 : 4816      DODU_FLG = ONE;
8241 : 4817      end;
8242 : 4818
8243 : 4819      ENDSUB;
8244 : 4820      end;
8245 : 4821
8246 : 4822      TST_PAT = not .TST_PAT;          !REPEAT WITH A 0 IN A FIELD OF 1'S
8247 : 4823      end;
8248 : 4824
8249 : 4825      TST_PAT = .TST_PAT^ONE;        !SHIFT THE 1 IN THE FIELD OF 0'S
8250 : 4826      end;
8251 : 4827
8252 : 4828      if .DODU_FLG IS_SET          !DROP THIS UNIT IF DODU_FLG IS SET
8253 : 4829      then
8254 : 4830      begin
8255 : 4831      DODU (.ML_LUN);
8256 : 4832      DOCLN;
8257 : 4833      end;
8258 : 4834
8259 : 4835      ENDTST;
8263 :

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (56)

8271	027660	004167	154232		ST6:	JSR	R1,\$SAVE5	:		4736
8272	027664	024646				CMP	-(SP),-(SP)	:		
8273	027666	005005				CLR	R5	:	DODU.FLG	4781
8274	027670	012703	0000C1			MOV	#1,R3	:	*,TST.PAT	4782
8275	027674	005001				CLR	R1	:	SHIFT	4784
8276	027676	005002			1\$:	CLR	R2	:	TWICE	4787
8277	027700	005004			2\$:	CLR	R4	:	REG.SEL	4790
8278	027702	104402			3\$:	TRAP	2	:		4791
8279	027704	152777	000040	163302		BISB	#40,@ML.REG+40	:		4792
8280	027712	016700	163664			MOV	ML,DUT,R0	:		
8281	027716	042700	177770			BIC	#177770,R0	:		
8282	027722	142777	000007	163264		BICB	#7,@ML.REG+40	:		
8283	027730	150077	163260			BISB	R0,@ML.REG+40	:		
8284	027734	010346				MOV	R3,-(SP)	:	TST.PAT,*	4794
8285	027736	010446				MOV	R4,-(SP)	:	REG.SEL,*	
8286	027740	012746	000010			MOV	#10,-(SP)	:		
8287	027744	060616				ADD	SP,(SP)	:	INDEX,*	
8288	027746	004767	173122			JSR	PC,WRT.REG	:		
8289	027752	010316				MOV	R3,(SP)	:	TST.PAT,*	4795
8290	027754	010446				MOV	R4,-(SP)	:	REG.SEL,*	
8291	027756	012746	000012			MOV	#12,-(SP)	:		
8292	027762	060616				ADD	SP,(SP)	:	ERR.FLG,*	
8293	027764	004767	173472			JSR	PC,RD.REG	:		
8294	027770	026627	000012	000001		CMP	12(SP),#1	:	ERR.FLG,*	4797
8295	027776	001106				BNE	7\$:		
8296	030000	005704				TST	R4	:	REG.SEL	4801
8297	030002	002410				BLT	4\$:		
8298	030004	020427	000003			CMP	R4,#3	:	REG.SEL,*	
8299	030010	003005				BGT	4\$:		
8300	030012	104455				TRAP	5\$:		4805
8301	030014	000005				.WORD	5	:		
8302	030016	010464				.WORD	ASYNCR	:		
8303	030020	024052				.WORD	DUMPER	:		
8304	030022	000425				BR	6\$:		4801
8305	030024	020427	000004		4\$:	CMP	R4,#4	:	REG.SEL,*	
8306	030030	002410				BLT	5\$:		
8307	030032	020427	000005			CMP	R4,#5	:	REG.SEL,*	
8308	030036	003005				BGT	5\$:		
8309	030040	104455				TRAP	5\$:		4808
8310	030042	000005				.WORD	5	:		
8311	030044	010526				.WORD	SYNCR	:		
8312	030046	024052				.WORD	DUMPER	:		
8313	030050	000412				BR	6\$:		4801
8314	030052	020427	000006		5\$:	CMP	R4,#6	:	REG.SEL,*	
8315	030056	002407				BLT	6\$:		
8316	030060	020427	000007			CMP	R4,#7	:	REG.SEL,*	
8317	030064	003004				BGT	6\$:		
8318	030066	104455				TRAP	5\$:		4811
8319	030070	000005				.WORD	5	:		
8320	030072	010570				.WORD	ARR.DAT	:		

```

8322      :ML4
8323      :
8324      :
8325 030074 024052
8326 030076 012746 007122      6$:      .WORD      DUMPER
8327 030102 012746 007066      MOV      #WORD.56,-(SP)      ;
8328 030106 012746 010004      MOV      #WORD.52,-(SP)
8329 030112 012746 007774      MOV      #FNC.6,-(SP)
8330 030116 012746 006426      MOV      #FNC.5,-(SP)
8331 030122 012746 007454      MOV      #WORD.12,-(SP)
8332 030126 012746 006100      MOV      #PHR.4,-(SP)
8333 030132 012746 000007      MOV      #SIX.FMT,-(SP)
8334 030136 010600      MOV      #7,-(SP)
8335 030140 104414      MOV      SP,R0      ; SP,*
8336 030142 016716 162762      TRAP     14
8337 030146 016746 162754      MOV      RD.DATA,(SP)      ;
8338 030152 016600 000036      MOV      WT.DATA,-(SP)      ;
8339 030156 006300      MOV      36(SP),R0      ; INDEX,*
8340 030160 006300      ASL     R0
8341 030162 006300      ASL     R0
8342 030164 016046 013154      MOV      ML.REG(R0),-(SP)
8343 030170 012746 005216      MOV      #FMT.16,-(SP)
8344 030174 012746 000004      MOV      #4,-(SP)
8345 030200 010600      MOV      SP,R0      ; SP,*
8346 030202 104414      TRAP     14
8347 030204 012705 000001      MOV      #1,R5      ; *,DODU.FLG
8348 030210 062706 000030      ADD     #30,SP      ;
8349 030214 062706 000012      7$:      ADD     #12,SP      ;
8350 030220 104467      TRAP     67      ;
8351 030222 006000      ROR     R0
8352 030224 103626      BLO     3$
8353 030226 005204      INC     R4      ; REG.SEL
8354 030230 020427 000007      CMP     R4,#7      ; REG.SEL,*
8355 030234 003622      BLE     3$
8356 030236 005103      COM     R3      ; TST.PAT
8357 030240 005202      INC     R2      ; TWICE
8358 030242 020227 000001      CMP     R2,#1      ; TWICE,*
8359 030246 003614      BLE     2$
8360 030250 006303      ASL     R3      ; TST.PAT
8361 030252 005201      INC     R1      ; SHIFT
8362 030254 020127 000017      CMP     R1,#17      ; SHIFT,*
8363 030260 003606      BLE     1$
8364 030262 005305      DEC     R5      ; DODU.FLG
8365 030264 001004      BNE     8$
8366 030266 016700 163306      MOV     ML.LUN,R0      ;
8367 030272 104451      TRAP     51
8368 030274 104444      TRAP     44
8369 030276 022626      8$:      CMP     (SP)+,(SP)+      ;
8370 030300 000207      RTS     PC
8371
8372      ; Routine Size: 137 words
8373      ; Maximum stack depth per invocation: 25 words

```


8382
8383
8387

8391 030302
8392 030302 004767 177352
8393 030306 104466
8394 030310 006000
8395 030312 103773
8396 030314 000207

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

4833

8397
8398
8399
8404
8405

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

8406 : 4836 !
8407 : 4837 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (57)

8409 :ML4
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 :
8444 :
8445 :
8446 :
8447 :
8448 :
8449 :
8450 :
8451 :
8452 :
8453 :
8454 :
8455 :
8456 :
8457 :
8458 :
8459 :
8460 :
8461 :
8462 :
8463 :

4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
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
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889

```

!
BGNTST:
!++
TEST NUMBER:  TST 7
TEST NAME:    REGISTER INITIALIZATION TEST
TEST DESCRIPTION:
THIS TEST TESTS THE ABILITY OF
ALL ACCESSIBLE ML11 REGISTERS
TO CLEAR OUT REGISTER DATA OF
ONE'S AND ZEROES PATTERN.

ROUTINE WRT REG WRITES A
PATTERN TO THE SELECTED REGISTER.

A MASS BUS CLEAR IS DONE.

THEN ROUTINE RD REG READS THE
SELECTED REGISTER FOR CLEARED DATA
THE DRIVE IS DROPPED ON DETECTED ERRORS.

THIS TEST WILL ALSO READ THE
DRIVE TYPE REGISTER FOR ITS
INITIAL REGISTER VALUE.

IMPLICIT INPUTS:
THIS FLAG IS NEEDED DUE TO THE UNIQUE
MANNER IN WHICH THESE REGISTERS
MUST BE READ.

RD DATA
LOADED BY THE READ REGISTER ROUTINE
AND CONTAINS THE DATA PATTERN READ FROM
THE REGISTER (REPRESENTS BAD DATA).

DRIVE TYPE
LOADED DURING THE INITIALIZATION CODE AND
STORES THE EXPECTED CONTENTS OF THE DRIVE
TYPE REGISTER.
    
```

```

local
TST_PAT,
ERR_FLG,
index,
CLR_DATA,
SAVE,
DODU_FLG;
    
```

```

!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
    
```

ML_REG [19, FORCE_HI] = %0'177777';

```

8465 :ML4
8466 :
8467 :
8468 : 4890 DODU_FLG = ZERO;
8469 : 4891 TST_PAT = ONES;
8470 : 4892
8471 : 4893 incr TWICE from 0 to 1 do
8472 : 4894 begin
8473 : 4895
8474 : 4896 incr REG_SEL from 0 to 10 do
8475 : 4897 begin
8476 : 4898 BGNSUB;
8477 : 4899 CLR_MBUS;
8478 : 4900 WRT_REG (.TST_PAT, .REG_SEL, index);
8479 : 4901 CLR_DATA = (.AI) or (.IGNORE);
8480 : 4902 CLR_MBUS;
8481 : 4903 RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);
8482 : 4904
8483 : 4905 if .ERR_FLG IS_SET
8484 : 4906 then
8485 : 4907 begin
8486 : 4908
8487 : 4909 selectone .REG_SEL of
8488 : 4910 set
8489 : 4911
8490 : 4912 [0, 1, 2, 3, 9, 10] :
8491 : 4913 ERRDF (6, ASYNC, DUMPER);
8492 : 4914
8493 : 4915 [4, 5, 8] :
8494 : 4916 ERRDF (6, SYNC, DUMPER);
8495 : 4917
8496 : 4918 [6 to 7] :
8497 : 4919 ERRDF (6, ARR_DAT, DUMPER);
8498 : 4920 tes;
8499 : 4921
8500 : 4922 PRINTB (FIV_FMT, PHR 4, WRD 12, WRD 52, FNC 23, WRD 56);
8501 : 4923 PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
8502 : 4924 DODU_FLG = ONE;
8503 : 4925 end;
8504 : 4926
8505 : 4927 ENDSUB;
8506 : 4928 end;
8507 : 4929
8508 : 4930 TST_PAT = not .TST_PAT;
8509 : 4931 end;
8510 : 4932
8511 : 4933 ML_REG [19, FORCE_HI] = ZEROES;
8512 : 4934
8513 : 4935
8514 : 4936 NOW TEST THE DRIVE TYPE REGISTER
8515 : 4937
8516 : 4938
8517 : 4939
8518 : 4940 if .MLDT neq .DRIVE_TYPE
8519 : 4941 then

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (57)

!BACKGROUND PATTERN
!REPEAT LOOP TWICE
!TEST THIRTEEN REGISTERS
!WRITE REGISTER WITH BACKGROUND
!CALCULATE THE CLEARED DATA PATTERN
!CLEAR THE REGISTER
!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
!REPEAT WITH COMPLIMENT BACKGROUND PAT
!RESTORE MLPD FORCED_HI
!COMPARE REG CONTENTS TO CALCULATED VALUE

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (57)

```

8521 :ML4
8522 :
8523 :
8524 : 4942 begin
8525 : 4943 ERRDF (113, ASYNC, DUMPER);
8526 : 4944 PRINTB (TWO_FMT, REG 7, PHR 4);
8527 : 4945 PRINTB (FMT_2, .DRIVE_TYPE, .SAVE);
8528 : 4946 end;
8529 : 4947
8530 : 4948 if .DODU_FLG IS_SET
8531 : 4949 then
8532 : 4950 begin
8533 : 4951 DODU (.ML_LUN);
8534 : 4952 DOCLN;
8535 : 4953 end;
8536 : 4954
8537 : 4955 ENDTST;

```

!DROP THIS UNIT IF DODU_FLG IS_SET

8545	030316	004167	153574	\$T7:	JSR	R1, \$SAVES	:	4835
8546	030322	162706	000006		SUB	#6, SP	:	
8547	030326	012767	177777	163052	MOV	#-1, ML.REG+232	:	4889
8548	030334	005005			CLR	R5	:	4890
8549	030336	012702	177777		MOV	#-1, R2	:	4891
8550	030342	005001			CLR	R1	:	4893
8551	030344	005004		1\$:	CLR	R4	:	4896
8552	030346	104402		2\$:	TRAP	2	:	4897
8553	030350	152777	000040	162636	BISB	#40, ML.REG+40	:	4898
8554	030356	016703	163220		MOV	ML.DUT, R3	:	
8555	030362	042703	177770		BIC	#177770, R3	:	
8556	030366	142777	000007	162620	BICB	#7, ML.REG+40	:	
8557	030374	150377	162614		BISB	R3, ML.REG+40	:	
8558	030400	010246			MOV	R2, -(SP)	:	4900
8559	030402	010446			MOV	R4, -(SP)	:	
8560	030404	012746	000012		MOV	#12, -(SP)	:	
8561	030410	060616			ADD	SP, (SP)	:	
8562	030412	004767	172456		JSR	PC, WRT.REG	:	
8563	030416	016600	000012		MOV	12(SP), R0	:	
8564	030422	006300			ASL	R0	:	
8565	030424	006300			ASL	R0	:	
8566	030426	006300			ASL	R0	:	
8567	030430	010003			MOV	R0, R3	:	
8568	030432	016300	013156		MOV	ML.REG+2(R3), R0	:	
8569	030436	056300	013162		BIS	ML.REG+6(R3), R0	:	
8570	030442	010066	000006		MOV	R0, 6(SP)	:	
8571	030446	152777	000040	162540	BISB	#40, ML.REG+40	:	
8572	030454	016700	163122		MOV	ML.DUT, R0	:	
8573	030460	042700	177770		BIC	#177770, R0	:	
8574	030464	142777	000007	162522	BICB	#7, ML.REG+40	:	

: *.CLR.DATA

Address	OpCode	Operand 1	Operand 2	Operand 3	Comments	Line No.
8576						
8577						
8578						
8579	030472	150077	162516		BISB R0,ML,REG+40	
8580	030476	016616	000006		MOV 6(SP),(SP)	; CLR.DATA,*
8581	030502	010446			MOV R4,-(SP)	; REG.SEL,*
8582	030504	012746	000014		MOV #14,-(SP)	
8583	030510	060616			ADD SP,(SP)	; ERR.FLG,*
8584	030512	004767	172744		JSR PC,RD,REG	
8585	030516	026627	000014	000001	CMP 14(SP),#1	; ERR.FLG,*
8586	030524	001110			BNE 10\$	
8587	030526	005704			TST R4	; REG.SEL
8588	030530	002403			BLT 3\$	
8589	030532	020427	000003		CMP R4,#3	; REG.SEL,*
8590	030536	003406			BLE 4\$	
8591	030540	020427	000011	3\$:	CMP R4,#11	; REG.SEL,*
8592	030544	002410			BLT 5\$	
8593	030546	020427	000012		CMP R4,#12	; REG.SEL,*
8594	030552	003005			BGT 5\$	
8595	030554	104455		4\$:	TRAP 55	
8596	030556	000006			.WORD 6	
8597	030560	010464			.WORD ASYNC	
8598	030562	024052			.WORD DUMPER	
8599	030564	000430			BR 9\$	
8600	030566	020427	000004	5\$:	CMP R4,#4	; REG.SEL,*
8601	030572	002403			BLT 6\$	
8602	030574	020427	000005		CMP R4,#5	; REG.SEL,*
8603	030600	003403			BLE 7\$	
8604	030602	020427	000010	6\$:	CMP R4,#10	; REG.SEL,*
8605	030606	001005			BNE 8\$	
8606	030610	104455		7\$:	TRAP 55	
8607	030612	000006			.WORD 6	
8608	030614	010526			.WORD SYNC	
8609	030616	024052			.WORD DUMPER	
8610	030620	000412			BR 9\$	
8611	030622	020427	000006	8\$:	CMP R4,#6	; REG.SEL,*
8612	030626	002407			BLT 9\$	
8613	030630	020427	000007		CMP R4,#7	; REG.SEL,*
8614	030634	003004			BGT 9\$	
8615	030636	104455			TRAP 55	
8616	030640	000006			.WORD 6	
8617	030642	010570			.WORD ARR.DAT	
8618	030644	024052			.WORD DUMPER	
8619	030646	012746	007122	9\$:	MOV #RD.56,-(SP)	
8620	030652	012746	010242		MOV #FNC.23,-(SP)	
8621	030656	012746	007066		MOV #RD.52,-(SP)	
8622	030662	012746	006426		MOV #RD.12,-(SP)	
8623	030666	012746	007454		MOV #PHR.4,-(SP)	
8624	030672	012746	006062		MOV #FIV.FMT,-(SP)	
8625	030676	012746	000006		MOV #6,-(SP)	
8626	030702	010600			MOV SP,R0	; SP,*
8627	030704	104414			TRAP 14	
8628	030706	016716	162216		MOV RD.DATA,(SP)	
8629	030712	016646	000030		MOV 30(SP),-(SP)	; CLR.DATA,*
8630	030716	016346	013154		MOV ML.REG(R3),-(SP)	

```

8632      ;ML4
8633      ;
8634
8635 030722 012746 005216      MOV      #FMT.6,-(SP)
8636 030726 012746 000004      MOV      #4,-(SP)
8637 030732 010600              MOV      SP,R0          ; SP,*
8638 030734 104414              TRAP     14
8639 030736 012705 000001      MOV      #1,R5          ; *,DODU.FLG
8640 030742 062706 000026      ADD      #26,SP          ;
8641 030746 062706 000012      ADD      #12,SP          ;
8642 030752 104467              TRAP     67             ;
8643 030754 006000              ROR      R0
8644 030756 103002              BHIS     12$
8645 030760 000167 177362      JMP      2$
8646 030764 005204              INC      R4              ; REG.SEL
8647 030766 020427 000012      CMP      R4,#12          ; REG.SEL,*
8648 030772 003772              BLE     11$
8649 030774 005102              COM      R2              ; TST.PAT
8650 030776 005201              INC      R1              ; TWICE
8651 031000 020127 000001      CMP      R1,#1          ; TWICE,*
8652 031004 003002              BGT     13$
8653 031006 000167 177332      JMP      1$
8654 031012 005067 162370      CLR      ML.REG+232      ;
8655 031016 027767 162262 162106  CMP      @ML.REG+130,DRIVE.TYPE ;
8656 031024 001431              BEQ     14$
8657 031026 104455              TRAP     55             ;
8658 031030 000161              .WORD   161
8659 031032 010464              .WORD   ASYNC
8660 031034 024052              .WORD   DUMPER
8661 031036 012746 007454      MOV      #PHR.4,-(SP)    ;
8662 031042 012746 010324      MOV      #REG.7,-(SP)
8663 031046 012746 006024      MOV      #TWO.FMT,-(SP)
8664 031052 012746 000003      MOV      #3,-(SP)
8665 031056 010600              MOV      SP,R0          ; SP,*
8666 031060 104414              TRAP     14
8667 031062 010416              MOV      R4,(SP)        ; SAVE,*
8668 031064 016746 162042      MOV      DRIVE.TYPE,-(SP)
8669 031070 012746 004266      MOV      #FMT.2,-(SP)
8670 031074 012746 000003      MOV      #3,-(SP)
8671 031100 010600              MOV      SP,R0          ; SP,*
8672 031102 104414              TRAP     14
8673 031104 062706 000016      ADD      #16,SP          ;
8674 031110 005305 14$:      DEC      R5              ; DODU.FLG
8675 031112 001004              BNE     15$
8676 031114 016700 162460      MOV      ML.LUN,R0      ;
8677 031120 104451              TRAP     51
8678 031122 104444              TRAP     44
8679 031124 062706 000006 15$:      ADD      #6,SP          ;
8680 031130 000207              RTS      PC
8681
8682
8683
; Routine Size: 198 words
; Maximum stack depth per invocation: 25 words

```

8692
8693
8697
8701 031132
8702 031132 004767 177160
8703 031136 104466
8704 031140 006000
8705 031142 103773
8706 031144 000207
8707
8708
8709
8714
8715
8716 :

T7::
1\$: JSR PC,\$T7 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

4953

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

4956 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (58)

```

8718 :ML4
8719 :
8720 :
8721 : 4957 :
8722 : 4958 :
8723 : 4959 :
8724 : 4960 BGNTST;
8725 : 4961 :
8726 : 4962 :++
8727 : 4963 : TEST NUMBER: TST 8
8728 : 4964 :
8729 : 4965 : TEST NAME: REGISTER SELECTION TEST
8730 : 4966 :
8731 : 4967 : TEST DESCRIPTION:
8732 : 4968 : THIS TEST TESTS FOR UNIQUE REGISTER
8733 : 4969 : SELECTION BY FIRST WRITING A BACKGROUND
8734 : 4970 : PATTERN INTO ALL READ/WRITE REGISTERS
8735 : 4971 :
8736 : 4972 : IT THEN WRITES A COMPLIMENT
8737 : 4973 : BACKGROUND PATTERN INTO ONE REGISTER
8738 : 4974 : AND READS THE REMAINING UNWRITTEN
8739 : 4975 : REGISTERS FOR AN UNCHANGED BACKGROUND
8740 : 4976 : PAT
8741 : 4977 :
8742 : 4978 : THIS PROCEDURE IS REPEATED UNTIL ALL
8743 : 4979 : REGISTERS HAVE BEEN WRITTEN WITH A
8744 : 4980 : COMPLIMENT BACKGROUND PATTERN.
8745 : 4981 :
8746 : 4982 :--
8747 : 4983 :
8748 : 4984 Local
8749 : 4985 DODU_FLG,
8750 : 4986 TST_PAT,
8751 : 4987 ERR_FLG,
8752 : 4988 index;
8753 : 4989
8754 : 4990 BGNSUB;
8755 : 4991 CLR MBUS;
8756 : 4992 DODU_FLG = ZERO;
8757 : 4993 TST_PAT = %0'125252';
8758 : 4994 WRT_CS1 (.TST_PAT, 0);
8759 : 4995 WRT_ER (.TST_PAT, 6);
8760 : 4996 WRT_DA (.TST_PAT, 3);
8761 : 4997 WRT_PA (.TST_PAT, 8);
8762 : 4998 WRT_E1 (.TST_PAT, 13);
8763 : 4999 WRT_E2 (.TST_PAT, 14);
8764 : 5000
8765 : 5001 incr CNT_1 from 0 to 4 do
8766 : 5002 begin
8767 : 5003 TST_PAT = not .TST_PAT;
8768 : 5004
8769 : 5005 case .CNT_1 from 0 to 4 of
8770 : 5006 set
8771 : 5007
8772 : 5008 [0] :

```

!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


```

8774 :ML4
8775 :
8776 :
8777 :      5009      WRT_CS1 (.TST_PAT, 0);      !FIRST PASS WRITE COMP PAT TO MLCS1
8778 :      5010
8779 :      [1] :      WRT_ER (.TST_PAT, 6);      !SECOND PASS WRITE COMP PAT TO MLER
8780 :      5012
8781 :      5013
8782 :      [2] :      WRT_DA (.TST_PAT, 3);      !THIRD PASS WRITE COMP PAT TO MLDA
8783 :      5015
8784 :      5016
8785 :      [3] :      WRT_PA (.TST_PAT, 8);      !FORTH PASS WRITE COMP PAT TO MLPA
8786 :      5018
8787 :      5019
8788 :      [4] :      WRT_E1 (.TST_PAT, 13);      !FIFTH PASS WRITE COMP PAT TO MLE1
8789 :      5021
8790 :      tes;      5022
8791 :      5023
8792 :      5024      TST_PAT = not .TST_PAT;      !COMPLIMENT TST_PAT BACK TO BACKGROUND
8793 :      5025
8794 :      inc: .CNT_2 from .CNT_1 + 1 to 5 do      !NOW READ THE REMAINING UNWRITTEN REGISTERS FOR AN
8795 :      begin      !UNCHANGED BACKGROUND
8796 :      5028
8797 :      case .CNT_2 from 0 to 5 of      !SELECT THE REGISTER TO READ
8798 :      set
8799 :      5031
8800 :      [0] :      RD_CS1 (.TST_PAT, index = 0, ERR_FLG); !READ MLCS1
8801 :      5033
8802 :      [1] :      RD_ER (.TST_PAT, index = 6, ERR_FLG); !READ MLER
8803 :      5035
8804 :      [2] :      RD_DA (.TST_PAT, index = 3, ERR_FLG); !READ MLDA
8805 :      5037
8806 :      [3] :      RD_PA (.TST_PAT, index = 8, ERR_FLG); !READ MLPA
8807 :      5039
8808 :      [4] :      RD_E1 (.TST_PAT, index = 13, ERR_FLG); !READ MLE1
8809 :      5041
8810 :      [5] :      RD_E2 (.TST_PAT, index = 14, ERR_FLG); !READ MLE2
8811 :      5043
8812 :      tes;      5044
8813 :      5045
8814 :      5046
8815 :      5047
8816 :      5048
8817 :      5049
8818 :      5050
8819 :      if .ERR_FLG IS_SET      !SEE IF READ FOUND AN ERROR
8820 :      then      5051
8821 :      begin      5052
8822 :      5053
8823 :      selectone .CNT_2 of      5054
8824 :      set      5055
8825 :      5056
8826 :      [0 to 2] :      5057
8827 :      ERRDF (110, ASYNC, DUMPER); !ASYNC MODULE FAILURE      5058
8828 :      5059
      5060

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
 22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (58)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (58)

```

8830 :ML4
8831 :
8832 :
8833 :           5061           [3 to 5] :
8834 :           5062           ERRDF (110, ARR_DAT, DUMPER);           !ARRAY DATA MODULE FAILURE
8835 :           5063           tes:
8836 :           5064
8837 :           5065           PRINTB (THR_FMT, WRD 38, WRD 37, WRD 10);
8838 :           5066           PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
8839 :           5067           DODU_FLG = ONE;
8840 :           5068           end;
8841 :           5069
8842 :           5070           end;
8843 :           5071
8844 :           5072           end;
8845 :           5073
8846 :           5074           ENDSUB;
8847 :           5075
8848 :           5076           if .DODU_FLG IS_SE*           !DROP THIS UNIT IF DODU_FLG IS_SET
8849 :           5077           then
8850 :           5078           begin
8851 :           5079           DODU (.ML_LUN);
8852 :           5080           DOCLN;
8853 :           5081           end;
8854 :           5082
8855 :           5083           ENDTST;

```

8863	031146	004167	152744	\$T8:	JSR	R1,\$SAVES	:	4955
8864	031152	005746			TST	-(SP)	:	
8865	031154	104402		1\$:	TRAP	2	:	4988
8866	031156	152777	000040	162030	BISB	#40,.ML.REG+40	:	4990
8867	031164	016703	162412		MOV	ML.DUT,R3		
8868	031170	042703	177770		BIC	#177770,R3		
8869	031174	142777	000007	162012	BICB	#7,.ML.REG+40		
8870	031202	150377	162006		BISB	R3,.ML.REG+40		
8871	031206	005005			CLR	R5	: DODU.FLG	4992
8872	031210	012704	125252		MOV	#-52526,R4	: *.TST.PAT	4993
8873	031214	010446			MOV	R4,-(SP)	: TST.PAT,*	4994
8874	031216	005046			CLR	-(SP)		
8875	031220	004767	166236		JSR	PC,WRT.CS1		
8876	031224	010416			MOV	R4,(SP)	: TST.PAT,*	4995
8877	031226	012746	000006		MOV	#6,-(SP)		
8878	031232	004767	166402		JSR	PC,WRT.ER		
8879	031236	010416			MOV	R4,(SP)	: TST.PAT,*	4996
8880	031240	012746	000003		MOV	#3,-(SP)		
8881	031244	004767	166546		JSR	PC,WRT.DA		
8882	031250	010416			MOV	R4,(SP)	: TST.PAT,*	4997
8883	031252	012746	000010		MOV	#10,-(SP)		

```

8885      :ML4
8886      :
8887
8888 031256 00767 167070      JSR    PC,WRT.PA
8889 031262 00416              MOV    R4,(SP)          ; TST.PAT,*      4998
8890 031264 012746 000015      MOV    #15,-(SP)
8891 031270 004767 167264      JSR    PC,WRT.E1
8892 031274 010416              MOV    R4,(SP)          ; TST.PAT,*      4999
8893 031276 012746 000016      MOV    #16,-(SP)
8894 031302 004767 167460      JSR    PC,WRT.E2
8895 031306 005001              CLR    R1              ; CNT.1          5001
8896 031310 005104 2$:      COM    R4              ; TST.PAT          5003
8897 031312 010103              MOV    R1,R3          ; CNT.1,*        5005
8898 031314 006303              ASL    R3
8899 031316 066307 031322      ADD    3$(R3),PC
8900 031322 000012 3$:      .WORD 4$-3$
8901 031324 000024              .WORD 5$-3$
8902 031326 000040              .WORD 6$-3$
8903 031330 000054              .WORD 7$-3$
8904 031332 000070              .WORD 8$-3$
8905 031334 010446 4$:      MOV    R4,-(SP)        ; TST.PAT,*      5009
8906 031336 005046              CLR    -(SP)
8907 031340 004767 166116      JSR    PC,WRT.CS1
8908 031344 000427              BR     9$
8909 031346 010446 5$:      MOV    R4,-(SP)        ; TST.PAT,*      5005
8910 031350 012746 000006      MOV    #6,-(SP)        ; TST.PAT,*      5012
8911 031354 004767 166260      JSR    PC,WRT.ER
8912 031360 000421              BR     9$
8913 031362 010446 6$:      MOV    R4,-(SP)        ; TST.PAT,*      5005
8914 031364 012746 000003      MOV    #3,-(SP)        ; TST.PAT,*      5015
8915 031370 004767 166422      JSR    PC,WRT.DA
8916 031374 000413              BR     9$
8917 031376 010446 7$:      MOV    R4,-(SP)        ; TST.PAT,*      5005
8918 031400 012746 000010      MOV    #10,-(SP)       ; TST.PAT,*      5018
8919 031404 004767 166742      JSR    PC,WRT.PA
8920 031410 000005              BR     9$
8921 031412 010446 8$:      MOV    R4,-(SP)        ; TST.PAT,*      5005
8922 031414 012746 000015      MOV    #15,-(SP)       ; TST.PAT,*      5021
8923 031420 004767 167134      JSR    PC,WRT.E1
8924 031424 005104 9$:      COM    R4              ; TST.PAT          5024
8925 031426 010103              MOV    R1,R3          ; CNT.1,CNT.2    5026
8926 031430 000576              BR     22$
8927 031432 010300 10$:     MOV    R3,R0          ; CNT.2,*        5029
8928 031434 006300              ACR   R0
8929 031436 066007 031442      ADD    11$(R0),PC
8930 031442 000014 11$:     .WORD 12$-11$
8931 031444 000036              .WORD 13$-11$
8932 031446 000062              .WORD 14$-11$
8933 031450 000106              .WORD 15$-11$
8934 031452 000132              .WORD 16$-11$
8935 031454 000156              .WORD 17$-11$
8936 031456 010446 12$:     MOV    R4,-(SP)        ; TST.PAT,*      5033
8937 031460 005002              CLR    R2              ; INDEX
8938 031462 005046              CLR    -(SP)
8939 031464 012746 000030      MOV    #30,-(SP)

```


22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

8997      ;ML4
8998      ;
8999
9000 031712 104455      TRAP      55      ;
9001 031714 000156      .WORD    156      ;
9002 031716 010570      .WORD    ARR.DAT
9003 031720 024052      .WORD    DUMPER
9004 031722 012746 006406 20$: MOV      #WRD.10,-(SP) ;
9005 031726 012746 006700      MOV      #WRD.37,-(SP) ;
9006 031732 012746 006710      MOV      #WRD.38,-(SP) ;
9007 031736 012746 006034      MOV      #THR.FMT,-(SP) ;
9008 031742 012746 000004      MOV      #4,-(SP)
9009 031746 010600      MOV      SP,R0      ; SP,*
9010 031750 104414      TRAP      14
9011 031752 016716 161152      MOV      RD.DATA,(SP) ;
9012 031756 016746 161144      MOV      WT.DATA,-(SP) ;
9013 031762 010200      MOV      R2,R0      ; INDEX,*
9014 031764 006300      ASL      R0
9015 031766 006300      ASL      R0
9016 031770 006300      ASL      R0
9017 031772 016046 013154      MOV      ML.REG(R0),-(SP)
9018 031776 012746 005216      MOV      #FMT.16,-(SP)
9019 032002 012746 000004      MOV      #4,-(SP)
9020 032006 010600      MOV      SP,R0      ; SP,*
9021 032010 104414      TRAP      14
9022 032012 012705 000001      MOV      #1,R5      ; *,DODU.FLG
9023 032016 062706 000022      ADD      #22,SP      ;
9024 032022 062706 000006 21$: ADD      #6,SP      ;
9025 032026 005203 22$: INC      R3      ; CNT.2
9026 032030 020327 000005      CMP      R3,#5      ; CNT.2,*
9027 032034 003002      BGT      23$
9028 032036 000167 177370      JMP      10$
9029 032042 022626 23$: CMP      (SP)+,(SP)+ ;
9030 032044 005201      INC      R1      ; CNT.1
9031 032046 020127 000004      CMP      R1,#4      ; CNT.1,*
9032 032052 003002      BGT      24$
9033 032054 000167 177230      JMP      2$
9034 032060 062706 000016 24$: ADD      #16,SP      ;
9035 032064 104467      TRAP      67      ;
9036 032066 006000      ROR      R0
9037 032070 103002      BHIS    25$
9038 032072 000167 177056      JMP      1$
9039 032076 005305 25$: DEC      R5      ; DODU.FLG
9040 032100 001004      BNE      26$
9041 032102 016700 161472      MOV      ML.LUN,R0 ;
9042 032106 104451      TRAP      51
9043 032110 104444      TRAP      44
9044 032112 005726 26$: TST      (SP)+ ;
9045 032114 000207      RTS      PC ;
9046
9047
9048
; Routine Size: 244 words
; Maximum stack depth per invocation: 28 words

```

9057
9058
9062
9066
9067
9068
9069
9070
9071
9072
9073
9074
9079
9080
9081 ;

032116
032116 004767 177024
032122 104466
032124 006000
032126 103773
032130 000207

T8::
1\$: JSR PC,\$T8 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

5081

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

5084 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (59)

```

9083 :ML4
9084 :
9085 :
9086 : 5085 :
9087 : 5086 :
9088 : 5087 BGNTST;
9089 : 5088 :
9090 : 5089 :++
9091 : 5090 : TEST NUMBER: TST 9
9092 : 5091 :
9093 : 5092 : TEST NAME: PRINT SERIAL NUMBER
9094 : 5093 :
9095 : 5094 : TEST DESCRIPTION:
9096 : 5095 : UPON A YES RESPONSE TO THE
9097 : 5096 : SOFTWARE QUESTION 'PRINT SERIAL NO?
9098 : 5097 : PRINT OUT THE DRIVE UNDER TEST
9099 : 5098 : SERIAL NUMBER.
9100 : 5099 :
9101 : 5100 :--
9102 : 5101 :
9103 : 5102 external
9104 : 5103 PRSN; !LOCATION WHERE ANSWER TO SW QUESTION IS STORED
9105 : 5104 :
9106 : 5105 local
9107 : 5106 D3; !STORES DIGIT 3 OF SERIAL NUMBER
9108 : 5107 D2; !STORES DIGIT 2 OF SERIAL NUMBER
9109 : 5108 D1; !STORES DIGIT 1 OF SERIAL NUMBER
9110 : 5109 D0; !STORES DIGIT 0 OF SERIAL NUMBER
9111 : 5110 :
9112 : 5111 if .PRSN IS_SET
9113 : 5112 then
9114 : 5113 begin
9115 : 5114 D3 = .SN3; !PRINT THE SERIAL NUMBER IF THE REPLY WAS YES
9116 : 5115 D2 = .SN2; !LOAD DIGIT 3 OF SN INTO D3
9117 : 5116 D1 = .SN1; !LOAD DIGIT 2 OF SN INTO D2
9118 : 5117 D0 = .SN0; !LOAD DIGIT 1 OF SN INTO D1
9119 : 5118 : !LOAD DIGIT 0 OF SN INTO D0
9120 : 5119 if ((.D3 gtr 9) or (.D2 gtr 9) or (.D1 gtr 9) or (.D0 gtr 9))
9121 : 5120 then !SEE IF DIGITS ARE TO BIG FOR COVERSION
9122 : 5121 PRINTB (FMT_25, PHR_15, .MLSN) !PRINT OCTAL SN IF TO BIG
9123 : 5122 else
9124 : 5123 PRINTB (FMT_26, PHR_15, .D3, .D2, .D1, .D0); !ELSE PRINT DECIMAL SERIAL NUMBER
9125 : 5124 :
9126 : 5125 end;
9127 : 5126 :
9128 : 5127 :
9129 : 5128 ENDTST;
9133 :
9134 : .GLOBL PRSN
9135 :
9136 :

```


22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
9194 ;ML4
9195 ;
9196 ;
9197 ; Routine Size: 72 words
9198 ; Maximum stack depth per invocation: 11 words
9203
9204
9208
9212 032352 T9::
9213 032352 004767 177554 1$: JSR PC,$T9 ;
9214 032356 104466 TRAP 66
9215 032360 006000 ROR R0
9216 032362 103773 BLO 1$
9217 032364 000207 RTS PC
9218
9219 ; Routine Size: 6 words
9220 ; Maximum stack depth per invocation: 0 words
9225
9226
9227 ; 5129 !<BLF/PAGE>
```

5126

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (60)

```

9229 :ML4
9230 :
9231 :
9232 : 5130 :
9233 : 5131 :
9234 : 5132 : BGNTST;
9235 : 5133 :
9236 : 5134 : ++
9237 : 5135 : TEST NUMBER: TST 10
9238 : 5136 :
9239 : 5137 : TEST NAME: C-BUS PARITY TEST
9240 : 5138 :
9241 : 5139 : TEST DESCRIPTION:
9242 : 5140 : TEST THE CONTROL BUS PARITY
9243 : 5141 : DETECTION AND GENERATING BY:
9244 : 5142 :
9245 : 5143 : 1. WRITING BAD PARITY TO DEVICE
9246 : 5144 : AND TEST CPAR SET.
9247 : 5145 :
9248 : 5146 : 2. WRITING GOOD PARITY TO DEVICE
9249 : 5147 : AND TEST CPAR CLR.
9250 : 5148 :
9251 : 5149 : 3. READING DEVICE AND TEST GOOD
9252 : 5150 : PARITY GENERATION BY READING
9253 : 5151 : MCPE CLR.
9254 : 5152 :
9255 : 5153 :
9256 : 5154 :
9257 : 5155 : local
9258 : 5156 : SAVE, !TEMPORARY SAVE LOCATION
9259 : 5157 : TST_PAT; !TEST PATTERN
9260 : 5158 :
9261 : 5159 : if .PAR_DIS IS_NOT_SET !SEE IF PARITY IS DISABLED
9262 : 5160 : then
9263 : 5161 : begin !BEGIN IF PARITY IS ENABLE
9264 : 5162 : TST_PAT = %o'125252'; !ALTERNATING 1, 0 PATTERN
9265 : 5163 :
9266 : 5164 : incr TWICE from 1 to 2 do !REPEAT LOOP TWICE
9267 : 5165 : begin
9268 : 5166 : CLR_MBUS;
9269 : 5167 : PAT = ONE; !GENERATE EVEN PARITY BY SETTING THE PAT BIT
9270 : 5168 : MLDA = .TST_PAT; !WRITE BAD PARITY ON CONTROL BUS
9271 : 5169 :
9272 : 5170 : if .CPAR IS_NOT_SET !READ CPAR BIT SET
9273 : 5171 : then
9274 : 5172 : begin
9275 : 5173 : ERRDF (7, ASYNC, DUMPER); !ERROR IF NOT SET
9276 : 5174 : PRINTB (THR_FMT, WRD_5, WRD_7, WRD_9);
9277 : 5175 : end;
9278 : 5176 :
9279 : 5177 : CLR_MBUS; !CLEAR OUT PAT BIT
9280 : 5178 : MLDA = .TST_PAT; !WRITE ODD PARITY CONTROL BUS
9281 : 5179 :
9282 : 5180 : if .CPAR IS_SET !READ CPAR BIT CLEARED
9283 : 5181 : then

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (60)

```

9285 :ML4
9286 :
9287 :
9288 :          5182
9289 :          5183      begin
9290 :          5184      ERRDF (8, ASYNC, DUMPER);          !ERROR IF SET
9291 :          5185      PRINTB (THR_FMT, WRD_6, WRD_7, WRD_9);
9292 :          5186      end;
9293 :          5187
9294 :          5188      CLR MBUS;
9295 :          5189      SAVE = .MLDA;          !READ A REGISTER FROM DUT
9296 :          5190
9297 :          5191      if .MCPE IS_SET          !SEE IF GENERATED GOOD PARITY
9298 :          5192      then
9299 :          5193      begin
9300 :          5194      ERRDF (9, ASYNC, DUMPER);          !ERROR IF MCPE IS_SET
9301 :          5195      PRINTB (THR_FMT, WRD_6, WRD_7, WRD_8);
9302 :          5196      end;
9303 :          5197
9304 :          5198      TST_PAT = .TST_PAT^ONE;          !REPEAT WITH SHIFTED DATA
9305 :          5199      end;
9306 :          5200      end
9307 :          5201      else
9308 :          5202      PRINTB (TWO_FMT, WRD_7, WRD_53);          !JUST PRINT MESSAGE IF PARITY IS DISABLED
9309 :          5203
9310 :          5204      ENDTST;
9311 :
9312 :
9313 :
9314 :
9318 032366 004167 151466      $T10: JSR      R1,$SAVE3          :          5128
9319 032372 005767 157170      TST      PAR.DIS          :          5159
9320 032376 001171          BNE      $S          :
9321 032400 012701 125252      MOV      #52526,R1          : *,TST.PAT          5162
9322 032404 012702 000001      MOV      #1,R2          : *,TWICE          5164
9323 032410 152777 000040 160576 1$: BISB     #40,@ML.REG+40          :          5165
9324 032416 016700 161160      MOV      ML,DUT,RO          :
9325 032422 042700 177770      BIC      #177770,RO          :
9326 032426 142777 000007 160560      BICB     #7,@ML.REG+40          :
9327 032434 150077 160554      BISB     RO,@ML.REG+40          :
9328 032440 152777 000020 160546      BISB     #20,@ML.REG+40          :
9329 032446 010177 160532      MOV      R1,@ML.REG+30          : TST.PAT,*          5167
9330 032452 132777 000010 160554      BITB     #10,@ML.REG+60          :          5168
9331 032460 001022          BNE      $S          :          5170
9332 032462 104455          TRAP     $S          :          5173
9333 032464 000007          .WORD    7          :
9334 032466 010464          .WORD    ASYNC          :
9335 032470 024052          .WORD    DUMPER          :
9336 032472 012746 006374      MOV      #WRD.9,-(SP)          :          5174
9337 032476 012746 006344      MOV      #WRD.7,-(SP)          :
9338 032502 012746 006330      MOV      #WRD.5,-(SP)          :

```

Address	OpCode	Operand 1	Operand 2	Operand 3	Comment	Label	Line No.
9340							
9341							
9342							
9343	032506	012746	006034		MOV #THR.FMT,-(SP)		
9344	032512	012746	000004		MOV #4,-(SP)		
9345	032516	010600			MOV SP,R0	: SP,*	
9346	032520	104414			TRAP 14	:	
9347	032522	062706	000012		ADD #12,SP	:	5172
9348	032526	152777	000040	160460	BISB #40,@ML.REG+40	:	5175
9349	032534	016700	161042		MOV ML,DUT,R0		
9350	032540	042700	177770		BIC #177770,R0		
9351	032544	142777	000007	160442	BICB #7,@ML.REG+40		
9352	032552	150077	160436		BISB R0,@ML.REG+40		
9353	032556	010177	160422		MOV R1,@ML.REG+30	: TST.PAT,*	5178
9354	032562	132777	000010	160444	BITB #10,@ML.REG+60	:	5180
9355	032570	001422			BEQ 3\$		
9356	032572	104455			TRAP 55	:	5183
9357	032574	000010			.WORD 10		
9358	032576	010464			.WORD ASYNC		
9359	032600	024052			.WORD DUMPER		
9360	032602	012746	006374		MOV #WRD.9,-(SP)	:	5184
9361	032606	012746	006344		MOV #WRD.7,-(SP)		
9362	032612	012746	006336		MOV #WRD.6,-(SP)		
9363	032616	012746	006034		MOV #THR.FMT,-(SP)		
9364	032622	012746	000004		MOV #4,-(SP)		
9365	032626	010600			MOV SP,R0	: SP,*	
9366	032630	104414			TRAP 14	:	
9367	032632	062706	000012		ADD #12,SP	:	5182
9368	032636	152777	000040	160350	BISB #40,@ML.REG+40	:	5185
9369	032644	016700	160732		MOV ML,DUT,R0		
9370	032650	042700	177770		BIC #177770,R0		
9371	032654	142777	000007	160332	BICB #7,@ML.REG+40		
9372	032662	150077	160326		BISB R0,@ML.REG+40		
9373	032666	017703	160312		MOV @ML.REG+30,R3	: *,SAVE	5188
9374	032672	032777	020000	160254	BIT #20000,@ML.REG	:	5190
9375	032700	001422			BEQ 4\$		
9376	032702	104455			TRAP 55	:	5193
9377	032704	000011			.WORD 11		
9378	032706	010464			.WORD ASYNC		
9379	032710	024052			.WORD DUMPER		
9380	032712	012746	006360		MOV #WRD.8,-(SP)	:	5194
9381	032716	012746	006344		MOV #WRD.7,-(SP)		
9382	032722	012746	006336		MOV #WRD.6,-(SP)		
9383	032726	012746	006034		MOV #THR.FMT,-(SP)		
9384	032732	012746	000004		MOV #4,-(SP)		
9385	032736	010600			MOV SP,R0	: SP,*	
9386	032740	104414			TRAP 14	:	
9387	032742	062706	000012		ADD #12,SP	:	5192
9388	032746	006301		4\$:	ASL R1	: TST.PAT	5197
9389	032750	005202			INC R2	: TWICE	5164
9390	032752	020227	000002		CMP R2,#2	: TWICE,*	
9391	032756	003614			BLE 1\$		
9392	032760	000207			RTS PC	:	5159
9393	032762	012746	007074	5\$:	MOV #WRD.53,-(SP)	:	5202
9394	032766	012746	006344		MOV #WRD.7,-(SP)	:	

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
9396 ;ML4
9397 ;
9398
9399 032772 012746 006024      MOV    #TWO.FMT,-(SP)
9400 032776 012746 000003      MOV    #3,-(SP)
9401 033002 010600              MOV    SP,R0          ; SP,*
9402 033004 104414              TRAP   14
9403 033006 062706 000010      ADD    #10,SP
9404 033012 000207              RTS    PC
9405
9406 ; Routine Size: 139 words
9407 ; Maximum stack depth per invocation: 9 words
9412
9413
9417
9421 033014
9422 033014 004767 177346      T10:: JSR    PC,$T10
9423 033020 104466              1$:   TRAP   66
9424 033022 006000              ROR    R0
9425 033024 103773              BLO   1$
9426 033026 000207              RTS    PC
9427
9428 ; Routine Size: 6 words
9429 ; Maximum stack depth per invocation: 0 words
9434
9435
9436 ;          5205 !<BLF/PAGE>
```

5128

5202

22-Dec-1980 09:24:31
 22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
 PA: <NEALE>BL3ML4.BLI.2 (61)

```

9438 :ML4
9439 :
9440 :
9441 : 5206 :
9442 : 5207 :
9443 : 5208 : BGNTST;
9444 : 5209 :
9445 : 5210 : ++
9446 : 5211 : TEST NUMBER: TST 11
9447 : 5212 :
9448 : 5213 : TEST NAME: MEMORY SIZING TEST
9449 : 5214 :
9450 : 5215 : TEST DESCRIPTION:
9451 : 5216 :
9452 : 5217 : THIS TESTS THE ML11'S SIZING
9453 : 5218 : LOGIC BY COMPARING THE
9454 : 5219 : OPERATORS INPUTED NUMBER OF ARRAYS
9455 : 5220 : PRESENT TO THE ML11 SIZING
9456 : 5221 : NUMBER OF ARRAYS PRESENT
9457 : 5222 : THE DRIVE IS DROPPED ON DETECTED ERRORS.
9458 : 5223 : IMPLICIT INPUTS:
9459 : 5224 :
9460 : 5225 : OP_NUM_ARR:
9461 : 5226 : LOADED DURING INIT CODE AND
9462 : 5227 : STORES OPERATORS INPUT TO THE
9463 : 5228 : SOFTWARE QUESTION:
9464 : 5229 : NUMBER OF ARRAYS PRESENT?
9465 : 5230 :
9466 : 5231 :
9467 : 5232 :
9468 : 5233 :
9469 : 5234 : Local
9470 : 5235 : DODU_FLG; !DROP UNIT FLAG
9471 : 5236 :
9472 : 5237 : BGNSUB;
9473 : 5238 : CLR MBUS;
9474 : 5239 : DODO_FLG = ZERO;
9475 : 5240 :
9476 : 5241 : if (.OP_NUM_ARR + 1) neq .ML_NUM_ARR !SEE IF DRIVE SIZED SAME NO. OF ARRAYS AS OP INPUTED
9477 : 5242 : then
9478 : 5243 : begin
9479 : 5244 : ERRORF (10, ASYNC, DUMPER); !IF NOT EQL THEN ERROR AND SET DODU_FLG
9480 : 5245 : PRINTB (TWO_FMT, FNC 1, WRD 14);
9481 : 5246 : PRINTB (FMT 2, (.OP_NUM_ARR + 1), .ML_NUM_ARR);
9482 : 5247 : DODU_FLG = ONE;
9483 : 5248 : end;
9484 : 5249 :
9485 : 5250 : ENDSUB;
9486 : 5251 :
9487 : 5252 : if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
9488 : 5253 : then
9489 : 5254 : begin
9490 : 5255 : DODU (.ML_LUN);
9491 : 5256 : DOCLN;
9492 : 5257 : end;
  
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (61)

```

9494 :ML4
9495 :
9496 :
9497 :      5258
9498 :      5259  ENDTST:
9502
9506 033030 004167 151024      $T11:  JSR      R1,SSAVE3
9507 033034 104402      1$:  TRAP     2
9508 033036 152777 000040 160150  B1SB    #40,@ML.REG+40
9509 033044 016701 160532      MOV     ML,DUT,R1
9510 033050 042701 177770      BIC     #177770,R1
9511 033054 142777 C.0007 160132  B1CB    #7,@ML.REG+40
9512 033062 150177 160126  B1SB    R1,@ML.REG+40
9513 033066 005002      CLR     R2
9514 033070 016701 156464      MOV     OP,NUM.ARR,R1
9515 033074 005201      INC     R1
9516 033076 017703 160172      MOV     @ML.REG+120,R3
9517 033102 006203      ASR    R3
9518 033104 006203      ASR    R3
9519 033106 006203      ASR    R3
9520 033110 000303      SWAB   R3
9521 033112 042703 177740      BIC     #177740,R3
9522 033116 010100      MOV     R1,R0
9523 033120 020003      CMP    R0,R3
9524 033122 001444      BEQ    2$
9525 033124 104455      TRAP   55
9526 033126 000012      .WORD  12
9527 033130 010464      .WORD  ASYNC
9528 033132 024052      .WORD  DUMPER
9529 033134 012746 006442      MOV     #WORD.14,-(SP)
9530 033140 012746 007726      MOV     #FNC.1,-(SP)
9531 033144 012746 006024      MOV     #TWO.FMT,-(SP)
9532 033150 012746 000003      MOV     #3,-(SP)
9533 033154 010600      MOV     SP,R0
9534 033156 104414      TRAP   14
9535 033160 017703 160110      MOV     @ML.REG+120,R3
9536 033164 006203      ASR    R3
9537 033166 006203      ASR    R3
9538 033170 006203      ASR    R3
9539 033172 000303      SWAB   R3
9540 033174 042703 177740      BIC     #177740,R3
9541 033200 010316      MOV     R3,(SP)
9542 033202 016746 156352      MOV     OP,NUM.ARR,-(SP)
9543 033206 005216      INC    (SP)
9544 033210 012746 004266      MOV     #FMT.2,-(SP)
9545 033214 012746 000003      MOV     #3,-(SP)
9546 033220 010600      MOV     SP,R0
9547 033222 104414      TRAP   14
    
```

```

9549      :ML4
9550      :
9551
9552 033224 012702 000001      MOV    #1,R2      ; *,DODU.FLG      5247
9553 033230 062706 000016      ADD    #16,SP    ;                   ;                   5243
9554 033234 104467              2$: TRAP  67      ;                   ;                   5248
9555 033236 006000              ROR    R0
9556 033240 103675              BLO   1$
9557 033242 005302              DEC   R2          ; DODU.FLG      5252
9558 033244 001004              BNE   3$
9559 033246 016700 160326      MOV    ML,LUN,R0 ;                   ;                   5255
9560 033252 104451              TRAP  51
9561 033254 104444              TRAP  44
9562 033256 000207              3$: RTS    PC    ;                   ;                   5204
9563
9564      : Routine Size: 76 words
9565      : Maximum stack depth per invocation: 11 words
9570
9571
9575
9579 033260              T11::
9580 033260 004767 177544      1$: JSR    PC,$T11 ;                   ;                   5257
9581 033264 104466              TRAP  66
9582 033266 006000              ROR    R0
9583 033270 103773              BLO   1$
9584 033272 000207              RTS    PC
9585
9586      : Routine Size: 6 words
9587      : Maximum stack depth per invocation: 0 words
9592
9593
9594 :           5260 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (62)

```

9596 :ML4
9597 :
9598 :
9599 :      5261 :
9600 :      5262 :
9601 :      5263 : BGNTST;
9602 :      5264 :
9603 :      5265 : !++
9604 :      5266 : TEST NUMBER: TST 12
9605 :      5267 :
9606 :      5268 : TEST NAME: NO-OP FUNCTION TEST
9607 :      5269 :
9608 :      5270 : TEST DESCRIPTION:
9609 :      5271 :
9610 :      5272 : TEST IF THE DRIVE CAN PERFORM
9611 :      5273 : A NO OP FUNCTION WITH OUT
9612 :      5274 : HANGING THE DRIVE.
9613 :      5275 :
9614 :      5276 : A NO_OP FUNCTION IS WRITTEN
9615 :      5277 : INTO MLCS1.
9616 :      5278 : THEN GO AND ERROR BITS ARE
9617 :      5279 : CHECKED FOR CORRECT STATES.
9618 :      5280 :
9619 :      5281 : THIS UNIT IS DROPPED ON DETECTED
9620 :      5282 : ERRORS.
9621 :      5283 :
9622 :      5284 : IMPLICIT INPUTS: NONE
9623 :      5285 :
9624 :      5286 :
9625 :      5287 :
9626 :      5288 :
9627 :      5289 : local
9628 :      5290 : DODU_FLG; !DROP UNIT FLAG
9629 :      5291 :
9630 :      5292 : BGNSUB;
9631 :      5293 : CLR MBUS;
9632 :      5294 : DODU_FLG = ZERO;
9633 :      5295 : MLCST = NOOP; !DO A NOOP FUNCTION
9634 :      5296 : DELAY (ONE_US); !DELAY 1 US
9635 :      5297 :
9636 :      5298 : if .GO IS_SET !SEE IF GO STILL SET
9637 :      5299 : then
9638 :      5300 : begin
9639 :      5301 : ERRDF (11, ASYNC, DUMPER); !ERROR AND SET DODU_FLG IF STILL SET
9640 :      5302 : PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_2, WRD_19);
9641 :      5303 : DODU_FLG = ONE;
9642 :      5304 : end;
9643 :      5305 :
9644 :      5306 : if .ILF IS_SET !SEE ILF SET
9645 :      5307 : then
9646 :      5308 : begin
9647 :      5309 : ERRDF (12, ASYNC, DUMPER); !ERROR AND SET DODU_FLG IF SET
9648 :      5310 : PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_2, WRD_19);
9649 :      5311 : DODU_FLG = ONE;
9650 :      5312 : end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (62)

```

9652 :ML4
9653 :
9654 :
9655 : 5313
9656 : 5314 if .OPI IS_SET !SEE IF CPI SET
9657 : 5315 then
9658 : 5316 begin
9659 : 5317 ERRDF (13, ASYNC, DUMPER); !ERROR AND SET DODU_FLG IF SET
9660 : 5318 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_11, FNC_2, WRD_19);
9661 : 5319 DODU_FLG = ONE;
9662 : 5320 end;
9663 : 5321
9664 : 5322 ENDSUB;
9665 : 5323
9666 : 5324 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
9667 : 5325 then
9668 : 5326 begin
9669 : 5327 DODU (.ML_LUN);
9670 : 5328 DOCLN;
9671 : 5329 end;
9672 : 5330
9673 : 5331 ENDTST;
9677 :

```

```

9681 033274 004167 150544 ST12: JSR R1,SSAVE2 ; 5259
9682 033300 005746 TST -(SP) ;
9683 033302 104402 1S: TRAP 2 ; 5290
9684 033304 152777 000040 157702 BISB #40,2ML.REG+40 ; 5292
9685 033312 016701 160264 MOV ML,DUT,R1
9686 033316 042701 177770 BIC #177770,R1
9687 033322 142777 000007 157664 BICB #7,2ML.REG+40
9688 033330 150177 157660 BISB R1,2ML.REG+40
9689 033334 055002 CLR R2 ; DODU.FLG 5294
9690 033336 057777 000001 157610 MOV #1,2ML.REG ; 5295
9691 033344 C 2700 000001 MOV #1,R0 ; *,SSTMP2 5296
9692 033350 001410 2S: BEQ 5S ;
9693 033352 016701 146540 MOV LSDLY,R1 ; *,SSTMP1
9694 033356 001403 BEQ 4S ;
9695 033360 005016 3S: CLR (SP) ; SSTMP
9696 033362 005301 DEC R1 ; SSTMP1
9697 033364 001375 BNE 3S ;
9698 033366 005300 4S: DEC R0 ; SSTMP2
9699 033370 000767 BR 2S ;
9700 033372 132777 000001 157554 5S: BITB #1,2ML.REG ; 5298
9701 033400 001430 BEQ 6S ;
9702 033402 104455 TRAP 5S ; 5301
9703 033404 000013 .WORD 13
9704 033406 010464 .WORD ASYNC
9705 033410 024052 .WORD DUMPER

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

9707          :ML4
9708          :
9709
9710 033412 012746 006506      MOV      #WORD.19,-(SP)      ;
9711 033416 012746 007742      MOV      #FNC.2,-(SP)      ;
9712 033422 012746 006416      MOV      #WORD.11,-(SP)   ;
9713 033426 012746 007404      MOV      #PHR.2,-(SP)     ;
9714 033432 012746 006276      MOV      #WORD.1,-(SP)   ;
9715 033436 012746 006062      MOV      #FIV.FMT,-(SP)  ;
9716 033442 012746 000006      MOV      #6,-(SP)        ;
9717 033446 010600              MOV      SP,R0            ; SP,*
9718 033450 104414              TRAP     14
9719 033452 012702 000001      MOV      #1,R2           ; *,DODU.FLG
9720 033456 062706 000016      ADD      #16,SP          ;
9721 033462 132777 000001 157544 6$: BITB     #1,@ML.REG+60    ;
9722 033470 001430              BEQ      7$              ;
9723 033472 104455              TRAP     55              ;
9724 033474 000014              .WORD   14
9725 033476 010464              .WORD   ASYNC
9726 033500 024052              .WORD   DUMPER
9727 033502 012746 006506      MOV      #WORD.19,-(SP)   ;
9728 033506 012746 007742      MOV      #FNC.2,-(SP)     ;
9729 033512 012746 006416      MOV      #WORD.11,-(SP)   ;
9730 033516 012746 007472      MOV      #PHR.5,-(SP)     ;
9731 033522 012746 006314      MOV      #WORD.3,-(SP)   ;
9732 033526 012746 006062      MOV      #FIV.FMT,-(SP)  ;
9733 033532 012746 000006      MOV      #6,-(SP)        ;
9734 033536 010600              MOV      SP,R0            ; SP,*
9735 033540 104414              TRAP     14
9736 033542 012702 000001      MOV      #1,R2           ; *,DODU.FLG
9737 033546 062706 000016      ADD      #16,SP          ;
9738 033552 032777 020000 157454 7$: BIT      #20000,@ML.REG+60 ;
9739 033560 001430              BEQ      8$              ;
9740 033562 104455              TRAP     55              ;
9741 033564 000015              .WORD   15
9742 033566 010464              .WORD   ASYNC
9743 033570 024052              .WORD   DUMPER
9744 033572 012746 006506      MOV      #WORD.19,-(SP)   ;
9745 033576 012746 007742      MOV      #FNC.2,-(SP)     ;
9746 033602 012746 006416      MOV      #WORD.11,-(SP)   ;
9747 033606 012746 007472      MOV      #PHR.5,-(SP)     ;
9748 033612 012746 006322      MOV      #WORD.4,-(SP)   ;
9749 033616 012746 006062      MOV      #FIV.FMT,-(SP)  ;
9750 033622 012746 000006      MOV      #6,-(SP)        ;
9751 033626 010600              MOV      SP,R0            ; SP,*
9752 033630 104414              TRAP     14
9753 033632 012702 000001      MOV      #1,R2           ; *,DODU.FLG
9754 033636 062706 000016      ADD      #16,SP          ;
9755 033642 104467              TRAP     67              ;
9756 033644 006000              ROR      R0
9757 033646 103615              BLO     1$
9758 033650 005302              DEC     R2                ; DODU.FLG
9759 033652 001004              BNE     9$
9760 033654 016700 157720      MOV      ML.LUN,R0       ;
9761 033660 104451              TRAP     51

```

5302

5303

5300

5306

5309

5310

5311

5308

5314

5317

5318

5319

5316

5320

5324

5327

9763
9764
9765
9766 033662 104444
9767 033664 005726
9768 033666 000207
9769
9770
9771
9776
9777
9781
9785 033670
9786 033670 004767 177400
9787 033674 104466
9788 033676 006000
9789 033700 103773
9790 033702 000207
9791
9792
9793
9798
9799
9800 : 5332 !<BLF/PAGE>

:ML4
:
9\$: TRAP 44
TST (SP)+
RTS PC ;
; Routine Size: 126 words
; Maximum stack depth per invocation: 11 words
T12::
1\$: JSR PC,\$T12 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC
; Routine Size: 6 words
; Maximum stack depth per invocation: 0 words

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

5259

5329

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (63)

```

9802 :ML4
9803 :
9804 :
9805 : 5333 |
9806 : 5334 |
9807 : 5335 | BGNSTST;
9808 : 5336 |
9809 : 5337 | ++
9810 : 5338 | TEST NUMBER: TST 13
9811 : 5339 |
9812 : 5340 | TEST NAME: WRITE CHECK FUNCTION TEST
9813 : 5341 |
9814 : 5342 | TEST DESCRIPTION:
9815 : 5343 |
9816 : 5344 | TEST IF THE DRIVE CAN PERFORM
9817 : 5345 | A WRITE CHECK FUNCTION WITHOUT
9818 : 5346 | HANGING THE DRIVE.
9819 : 5347 |
9820 : 5348 | A WRITE CHECK FUNCTION IS WRITTEN
9821 : 5349 | INTO MLCS1.
9822 : 5350 | THEN GO AND ERROR BITS ARE
9823 : 5351 | CHECKED FOR CORRECT STATES
9824 : 5352 |
9825 : 5353 |
9826 : 5354 |
9827 : 5355 | local
9828 : 5356 | DODU_FLG; !DROP UNIT FLAG
9829 : 5357 |
9830 : 5358 | BGNSUB;
9831 : 5359 | CLR MBUS;
9832 : 5360 | DODU_FLG = ZERO;
9833 : 5361 | FIRST_BLK_XFER (); !SET UP A FIRST BLK XFERR
9834 : 5362 | MLCS1 = WRT_CHK; !DO A WRITE CHECK FUNCTION
9835 : 5363 |
9836 : 5364 | if .GO IS_NOT_SET !SEE IF THE GO BIT GOT SET
9837 : 5365 | then
9838 : 5366 | begin
9839 : 5367 | ERRDF (14, ASYNC, DUMPER); !ERROR IF NOT SET
9840 : 5368 | PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_4, WRD_19);
9841 : 5369 |
9842 : 5370 | if .DRY IS_NOT_SET !SEE IF THE DRY IS SET WHILE GO IS CLEARED
9843 : 5371 | then
9844 : 5372 | begin
9845 : 5373 | ERRDF (15, ASYNC, DUMPER); !ERROR IF CLEARED
9846 : 5374 | PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
9847 : 5375 | end;
9848 : 5376 |
9849 : 5377 | DODU_FLG = ONE; !SET DODU_FLG
9850 : 5378 | end
9851 : 5379 | else
9852 : 5380 |
9853 : 5381 | if .DRY IS_SET !THE GO IS SET SO SEE IF DRY IS CLEARED
9854 : 5382 | then
9855 : 5383 | begin
9856 : 5384 | ERRDF (16, ASYNC, DUMPER); !ERROR IF DRY IS SET

```

```

9858 :ML4
9859 :
9860 :
9861 : 5385 PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
9862 : 5386 DODU_FLG = ONE;
9863 : 5387 end;
9864 : 5388
9865 : 5389 if .ILF IS_SET !SEE IF ILF IS SET
9866 : 5390 then
9867 : 5391 begin
9868 : 5392 ERRDF (17, ASYNC, DUMPER); !ERROR IF SET
9869 : 5393 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_4, WRD_19);
9870 : 5394 DODU_FLG = ONE;
9871 : 5395 end;
9872 : 5396
9873 : 5397 if .OPI IS_SET !SEE IF OPI IS SET
9874 : 5398 then
9875 : 5399 begin
9876 : 5400 ERRDF (18, ASYNC, DUMPER); !ERROR IF SET
9877 : 5401 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_4, WRD_19);
9878 : 5402 DODU_FLG = ONE;
9879 : 5403 end;
9880 : 5404
9881 : 5405 DELAY (FRTY_US); !WAIT FO XFERR TO COMPLTE
9882 : 5406
9883 : 5407 if .DRY IS_NOT_SET !SEE IF DRY IS SETS AFTER XFERR
9884 : 5408 then
9885 : 5409 begin
9886 : 5410
9887 : 5411 if .GO IS_SET !IF DRY IS NOT SET THEN SEE IF GO IS SET
9888 : 5412 then
9889 : 5413 begin
9890 : 5414 CLR_MBUS; !IF THE GO IS SET THEN TRY TO CLR GO
9891 : 5415
9892 : 5416 if .GO IS_SET then ERRDF (19, ASYNC, DUMPER) else ERRDF (20, SYNC, DUMPER);
9893 : 5417
9894 : 5418 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19); !IF GO IS STILL SET THEN ASYNC FAILURE
9895 : 5419 !ELSE SYNC MODLE FAILURE
9896 : 5420 end
9897 : 5421 else
9898 : 5422 begin
9899 : 5423 ERRDF (21, ASYNC, DUMPER); !ERPOR GO AND DRY BOTH CLEARED
9900 : 5424 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_4, WRD_19);
9901 : 5425 end;
9902 : 5426
9903 : 5427 DODU_FLG = ONE; !SET THE DODU_FLG
9904 : 5428 end;
9905 : 5429
9906 : 5430 if .GO IS_SET !SEE IF THE GO IS STILL SET
9907 : 5431 then
9908 : 5432 begin
9909 : 5433 CLR_MBUS; !TRY TO CLR GO IF STILL SET
9910 : 5434
9911 : 5435 if .GO IS_SET then ERRDF (22, ASYNC, DUMPER) else ERRDF (23, SYNC, DUMPER);
9912 : 5436

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (63)

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:(NEALE)BL3ML4.BLI.2 (63)

```

9914 :ML4
9915 :
9916 :
9917 : 5437
9918 : 5438 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19); !IF GO IS STILL SET THE ASYNC FAILURE
9919 : 5439 DODU_FLG = ONE; !ELSE SYNC MODULE FAILURE
9920 : 5440 end;
9921 : 5441
9922 : 5442 ENDSUB;
9923 : 5443
9924 : 5444 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
9925 : 5445 then
9926 : 5446 begin
9927 : 5447 DODU (.ML_LUN);
9928 : 5448 DOCLN;
9929 : 5449 end;
9930 : 5450
9931 : 5451 ENDTST;
9935 :
9939 033704 004167 150134 $T13: JSR R1,$SAVE2 ; 5331
9940 033710 005746 TST -(SP) ;
9941 033712 104402 1$: TRAP 2 ; 5356
9942 033714 152777 000040 157272 BISB #40,@ML.REG+40 ; 5358
9943 033722 016701 157654 MOV ML,DUT,R1 ;
9944 033726 042701 177770 BIC #177770,R1 ;
9945 033732 142777 000007 157254 BICB #7,@ML.REG+40 ;
9946 033740 150177 157250 BISB R1,@ML.REG+40 ;
9947 033744 005002 CLR R2 ; DODU.FLG 5360
9948 033746 004767 162066 JSR PC,FIRST.BLK.XFER ; 5361
9949 033752 012777 000051 157174 MOV #51,@ML.REG ; 5362
9950 033760 132777 000001 157166 BITB #1,@ML.REG ; 5364
9951 033766 001057 BNE 2$ ;
9952 033770 104455 TRAP 55 ; 5367
9953 033772 000016 .WORD 16 ;
9954 033774 010464 .WORD ASYNC ;
9955 033776 024052 .WORD DUMPER ;
9956 034000 012746 006506 MOV #WRD.19,-(SP) ; 5368
9957 034004 012746 007756 MOV #FNC.4,-(SP) ;
9958 034010 012746 006426 MOV #WRD.12,-(SP) ;
9959 034014 012746 007366 MOV #PHR.1,-(SP) ;
9960 034020 012746 006276 MOV #WRD.1,-(SP) ;
9961 034024 012746 006062 MOV #FIV_FMT,-(SP) ;
9962 034030 012746 000006 MOV #6,-(SP) ;
9963 034034 010600 MOV SP,R0 ; SP,*
9964 034036 104414 TRAP 14 ;
9965 034040 132777 000200 157156 BITB #200,@ML.REG+50 ; 5370
9966 034046 001056 BNE 3$ ;
9967 034050 104455 TRAP 55 ; 5373

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

9969          :ML4
9970          :
9971
9972 034052 000017          .WORD 17
9973 034054 010464          .WORD ASYNC
9974 034056 024052          .WORD DUMPER
9975 034060 012746 007504  MOV #PHR.6,-(SP)          :
9976 034064 012746 006276  MOV #WRD.1,-(SP)          :
9977 034070 012746 006754  MOV #WRD.43,-(SP)         :
9978 034074 012746 007366  MOV #PHR.1,-(SP)         :
9979 034100 012746 006302  MOV #WRD.2,-(SP)         :
9980 034104 012746 006062  MOV #FIV.FMT,-(SP)      :
9981 034110 012746 000006  MOV #6,-(SP)           :
9982 034114 010600          MOV SP,R0                : SP,*
9983 034116 104414          TRAP 14                   :
9984 034120 062706 000016  ADD #16,SP               :
9985 034124 000427          BR 3$                    :
9986 034126 105777 157072 2$: TSTB @ML.REG+50          :
9987 034132 100030          BPL 4$                   :
9988 034134 104455          TRAP 55                  :
9989 034136 000020          .WORD 20                 :
9990 034140 010464          .WORD ASYNC
9991 034142 024052          .WORD DUMPER
9992 034144 012746 007472  MOV #PHR.5,-(SP)          :
9993 034150 012746 006276  MOV #WRD.1,-(SP)          :
9994 034154 012746 006754  MOV #WRD.43,-(SP)         :
9995 034160 012746 007404  MOV #PHR.2,-(SP)         :
9996 034164 012746 006302  MOV #WRD.2,-(SP)         :
9997 034170 012746 006062  MOV #FIV.FMT,-(SP)      :
9998 034174 012746 000006  MOV #6,-(SP)           :
9999 034200 010600          MOV SP,R0                : SP,*
10000 034202 104414          TRAP 14
10001 034204 012702 000001 3$: MOV #1,R2                : *,DODU.FLG
10002 034210 062706 000016  ADD #16,SP               :
10003 034214 132777 000001 157012 4$: BITB #1,@ML.REG+60      :
10004 034222 001430          BEQ 5$                   :
10005 034224 104455          TRAP 55                  :
10006 034226 000021          .WORD 21                 :
10007 034230 010464          .WORD ASYNC
10008 034232 024052          .WORD DUMPER
10009 034234 012746 006506  MOV #WRD.19,-(SP)        :
10010 034240 012746 007756  MOV #FNC.4,-(SP)         :
10011 034244 012746 006426  MOV #WRD.12,-(SP)        :
10012 034250 012746 007472  MOV #PHR.5,-(SP)         :
10013 034254 012746 006314  MOV #WRD.3,-(SP)         :
10014 034260 012746 006062  MOV #FIV.FMT,-(SP)      :
10015 034264 012746 000006  MOV #6,-(SP)           :
10016 034270 010600          MOV SP,R0                : SP,*
10017 034272 104414          TRAP 14
10018 034274 012702 000001  MOV #1,R2                : *,DODU.FLG
10019 034300 062706 000016  ADD #16,SP               :
10020 034304 032777 020000 156722 5$: BIT #20000,@ML.REG+60    :
10021 034312 001430          BEQ 6$                   :
10022 034314 104455          TRAP 55                  :
10023 034316 000022          .WORD 22

```


Address	Hex	Hex	Hex	Hex	Label	Code	Comment	Line
10025								
10026								
10027								
10028	034320	010464				.WORD	ASYN	
10029	034322	024052				.WORD	DUMPER	
10030	034324	012746	006506			MOV	#WORD.19,-(SP)	5401
10031	034330	012746	007756			MOV	#FNC.4,-(SP)	
10032	034334	012746	006426			MOV	#WORD.12,-(SP)	
10033	034340	012746	007472			MOV	#PHR.5,-(SP)	
10034	034344	012746	006322			MOV	#WORD.4,-(SP)	
10035	034350	012746	006062			MOV	#FIV.FMT,-(SP)	
10036	034354	012746	000006			MOV	#6,-(SP)	
10037	034360	010600				MOV	SP,R0	: SP,*
10038	034362	104414				TRAP	14	
10039	034364	012702	000001			MOV	#1,R2	: *,DODU.FLG 5402
10040	034370	062706	000016			ADD	#16,SP	: 5399
10041	034374	012700	000050			MOV	#50,R0	: *,SSTMP2 5405
10042	034400	001410			6\$:	BEQ	10\$	
10043	034402	016701	145510		7\$:	MOV	LSDLY,R1	: *,SSTMP1
10044	034406	001403				BEQ	9\$	
10045	034410	005016			8\$:	CLR	(SP)	: SSTMP
10046	034412	005301				DEC	R1	: SSTMP1
10047	034414	001375				BNE	8\$	
10048	034416	005300			9\$:	DEC	R0	: SSTMP2
10049	034420	000767				BR	7\$	
10050	034422	132777	000200	156574	10\$:	BITB	#200,@ML.REG+50	: 5407
10051	034430	001106				BNE	15\$	
10052	034432	132777	000001	156514		BITB	#1,@ML.REG	: 5411
10053	034440	001452				BEQ	13\$	
10054	034442	152777	000040	156544		BISB	#40,@ML.REG+40	: 5413
10055	034450	016701	157126			MOV	ML.DUT,R1	
10056	034454	042701	177770			BIC	#177770,R1	
10057	034460	142777	000007	156526		BICB	#7,@ML.REG+40	
10058	034466	150177	156522			BISB	R1,@ML.REG+40	
10059	034472	132777	000001	156454		BITB	#1,@ML.REG	: 5416
10060	034500	001405				BEQ	11\$	
10061	034502	104455				TRAP	55	
10062	034504	000023				.WORD	23	
10063	034506	010464				.WORD	ASYN	
10064	034510	024052				.WORD	DUMPER	
10065	034512	000404				BR	12\$	
10066	034514	104455			11\$:	TRAP	55	
10067	034516	000024				.WORD	24	
10068	034520	010526				.WORD	SYN	
10069	034522	024052				.WORD	DUMPER	
10070	034524	012746	006506		12\$:	MOV	#WORD.19,-(SP)	: 5419
10071	034530	012746	007756			MOV	#FNC.4,-(SP)	
10072	034534	012746	006416			MOV	#WORD.11,-(SP)	
10073	034540	012746	007404			MOV	#PHR.2,-(SP)	
10074	034544	012746	006276			MOV	#WORD.1,-(SP)	
10075	034550	012746	006062			MOV	#FIV.FMT,-(SP)	
10076	034554	012746	000006			MOV	#6,-(SP)	
10077	034560	010600				MOV	SP,R0	: SP,*
10078	034562	104414				TRAP	14	
10079	034564	000424				BR	14\$: 5411

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

10081										
10082										
10083										
10084	034566	104455		13\$:	TRAP	55				5423
10085	034570	000025			.WORD	25				
10086	034572	010464			.WORD	ASYN				
10087	034574	024052			.WORD	DUMPER				
10088	034576	012746	006506		MOV	#WRD.19,-(SP)				5424
10089	034602	012746	007756		MOV	#FNC.4,-(SP)				
10090	034606	012746	006416		MOV	#WRD.11,-(SP)				
10091	034612	012746	007366		MOV	#PHR.1,-(SP)				
10092	034616	012746	006302		MOV	#WRD.2,-(SP)				
10093	034622	012746	006062		MOV	#FIV.FMT,-(SP)				
10094	034626	012746	000006		MOV	#6,-(SP)				
10095	034632	010600			MOV	SP,R0				
10096	034634	104414			TRAP	14				
10097	034636	012702	000001	14\$:	MOV	#1,R2				5427
10098	034642	062706	000016		ADD	#16,SP				5409
10099	034646	132777	000001	156300	15\$:	BITB	#1,BPL.REG			5430
10100	034654	001455			BEQ	18\$				
10101	034656	152777	000040	156330	BISB	#40,BPL.REG+40				5432
10102	034664	016701	156712		MOV	ML,DUT,R1				
10103	034670	042701	177770		BIC	#177770,R1				
10104	034674	142777	000007	156312	BICB	#7,BPL.REG+40				
10105	034702	150177	156306		BISB	R1,BPL.REG+40				
10106	034706	132777	000001	156240	BITB	#1,BPL.REG				5435
10107	034714	001405			BEQ	16\$				
10108	034716	104455			TRAP	55				
10109	034720	000026			.WORD	26				
10110	034722	010464			.WORD	ASYN				
10111	034724	024052			.WORD	DUMPER				
10112	034726	000404			BR	17\$				
10113	034730	104455		16\$:	TRAP	55				
10114	034732	000027			.WORD	27				
10115	034734	010526			.WORD	SYN				
10116	034736	024052			.WORD	DUMPER				
10117	034740	012746	006506	17\$:	MOV	#WRD.19,-(SP)				5438
10118	034744	012746	007756		MOV	#FNC.4,-(SP)				
10119	034750	012746	006416		MOV	#WRD.11,-(SP)				
10120	034754	012746	007404		MOV	#PHR.2,-(SP)				
10121	034760	012746	006276		MOV	#WRD.1,-(SP)				
10122	034764	012746	006062		MOV	#FIV.FMT,-(SP)				
10123	034770	012746	000006		MOV	#6,-(SP)				
10124	034774	010600			MOV	SP,R0				
10125	034776	104414			TRAP	14				
10126	035000	012702	000001		MOV	#1,R2				5439
10127	035004	062706	000016		ADD	#16,SP				5432
10128	035010	104467		18\$:	TRAP	67				5440
10129	035012	006000			ROR	R0				
10130	035014	103002			BHIS	19\$				
10131	035016	000167	176670		JMP	1\$				
10132	035022	005302		19\$:	DEC	R2				5444
10133	035024	001004			BNE	20\$				
10134	035026	016700	156546		MOV	ML,LUN,R0				5447
10135	035032	104451			TRAP	51				

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
10137      :ML4
10138      :
10139
10140 035034 104444
10141 035036 005726
10142 035040 000207
10143
10144      : Routine Size: 303 words
10145      : Maximum stack depth per invocation: 18 words
10150
10151
10155
10159 035042
10160 035042 004767 176636
10161 035046 104466
10162 035050 006000
10163 035052 103773
10164 035054 000207
10165
10166      : Routine Size: 6 words
10167      : Maximum stack depth per invocation: 0 words
10172
10173
10174 :      5452 !<BLF/PAGE>
```

```
20$:      TRAP      44
          TST      (SP)+
          RTS      PC
```

5331

```
T13::
1$:      JSR      PC,$T13
          TRAP     66
          ROR     R0
          BLO    1$
          RTS     PC
```

5449

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (64)

```

10176 :ML4
10177 :
10178 :
10179 :      5453 !
10180 :      5454 BGNSTST;
10181 :      5455
10182 :      5456 !++
10183 :      5457 ! TEST NUMBER: TST 14
10184 :      5458
10185 :      5459 ! TEST NAME: WRITE FUNCTION TEST
10186 :      5460
10187 :      5461 ! TEST DESCRIPTION:
10188 :      5462 ! TEST IF THE DRIVE CAN PERFORM A WRITE FUNCTION WITHOUT
10189 :      5463 ! HANGING THE DRIVE.
10190 :      5464
10191 :      5465 ! A WRITE FUNCTION IS WRITTEN INTO MLCS1. THEN GO AND ERROR BITS ARE
10192 :      5466 ! CHECKED FOR CORRECT STATUS. THIS UNIT IS DROPPED ON DETECTED ERRORS.
10193 :      5467 !--
10194 :      5468
10195 :      5469 local
10196 :      5470 DODU_FLG; !DROP UNIT FLAG
10197 :      5471
10198 :      5472 BGNSUB;
10199 :      5473 CLR MBUS;
10200 :      5474 DODU_FLG = ZERO;
10201 :      5475 FIRST_BLK_XFER (); !SET UP A FIRST BLOCK XFERR
10202 :      5476 MLCS1 = write; !DO A WRITE FUNCTION
10203 :      5477
10204 :      5478 if .GO IS_NOT_SET !SEE IF THE GO IS SET
10205 :      5479 then
10206 :      5480 !ERROR IF NOT SET
10207 :      5481 !begin
10208 :      5482 ERRDF (24, ASYNC, DUMPER);
10209 :      5483 PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_5, WRD_19);
10210 :      5484
10211 :      5485 if .DRY IS_NOT_SET !SEE IF DRY SET WITH GO CLEAR
10212 :      5486 then
10213 :      5487 !begin
10214 :      5488 ERRDF (25, ASYNC, DUMPER);
10215 :      5489 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
10216 :      5490 !end;
10217 :      5491 DODU_FLG = ONE; !SET DODU_FLG
10218 :      5492 !end
10219 :      5493 else !GO IS SET DURING FUNCTION
10220 :      5494
10221 :      5495 if .DRY IS_SET !SEE IF DRY CLEAR WITH GO SET
10222 :      5496 then
10223 :      5497 !ERROR IF SET
10224 :      5498 !begin
10225 :      5499 ERRDF (26, ASYNC, DUMPER);
10226 :      5500 PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
10227 :      5501 DODU_FLG = ONE;
10228 :      5502 !end;
10229 :      5503 if .ILF IS_SET !DID FUNCTION CAUSE ILF
10230 :      5504 then

```

```

10232 :ML4
10233 :
10234 :
10235 : 5505 begin !ERROR IF YES
10236 : 5506 ERRDF (27, ASYNC, DUMPER);
10237 : 5507 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_5, WRD_19);
10238 : 5508 DODU_FLG = ONE;
10239 : 5509 end;
10240 :
10241 : 5510
10242 : 5511 if .OPI IS_SET !DID FUNCTION CAUSE OPI
10243 : 5512 then
10244 : 5513 begin !ERROR IF YES
10245 : 5514 ERRDF (28, ASYNC, DUMPER);
10246 : 5515 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_5, WRD_19);
10247 : 5516 DODU_FLG = ONE;
10248 : 5517 end;
10249 : 5518
10250 : 5519 DELAY (FRTY_US); !WAIT FOR XFERR TO COMPLETE
10251 : 5520
10252 : 5521 if .DRY IS_NOT_SET !SFE IF DRY CLEARED AFTER XFERR
10253 : 5522 then
10254 : 5523 begin
10255 : 5524
10256 : 5525 if .GO IS_SET !TST GO CLR IF DRY NOT SET
10257 : 5526 then
10258 : 5527 begin
10259 : 5528 CLR_MBUS; !CLEAR GO IF STILL SET
10260 : 5529
10261 : 5530 if .GO IS_SET then ERRDF (29, ASYNC, DUMPER) else ERRDF (30, SYNC, DUMPER);
10262 : 5531
10263 : 5532 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19); !TST GO TO DETERMINE FAILING MOD
10264 : 5533 end
10265 : 5534 else !DRY NOT SET AND GO CLEARED
10266 : 5535 begin !REPORT ERROR
10267 : 5536 ERRDF (31, ASYNC, DUMPER);
10268 : 5537 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_5, WRD_19);
10269 : 5538 end;
10270 : 5539
10271 : 5540
10272 : 5541 DODU_FLG = ONE; !SET DODU_FLG
10273 : 5542 end;
10274 : 5543
10275 : 5544 if .GO IS_SET !SEE IF GO CLEARED AFTER XFERR.
10276 : 5545 then
10277 : 5546 begin
10278 : 5547 CLR_MBUS; !CLEAR GO IF STILL SET
10279 : 5548
10280 : 5549 if .GO IS_SET then ERRDF (32, ASYNC, DUMPER) else ERRDF (33, SYNC, DUMPER);
10281 : 5550
10282 : 5551 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19); !TST GO TO DETERMINE FAILING MOD
10283 : 5552 DODU_FLG = ONE;
10284 : 5553 end;
10285 : 5554
10286 : 5555
10287 : 5556 ENDSUB;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BLJML4.BLI.2 (64)

22-Dec-1980 09:24:31 TOPS-20 BL1sg-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (64)

```
10288 :ML4
10289 :
10290 :
10291 : 5557
10292 : 5558 if .TRE IS_SET !SEE IF XFERR CAUSED A TRANSFER ERROR
10293 : 5559 then
10294 : 5560 begin !REPORT ERROR IF SET AND CONTINUE TESTING
10295 : 5561 ERADF (115, INTER, DUMPER);
10296 : 5562 PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_5, WRD_19);
10297 : 5563 DODU_FLG = ONE;
10298 : 5564 end;
10299 : 5565
10300 : 5566 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG SET
10301 : 5567 then
10302 : 5568 begin
10303 : 5569 DODU (.ML_LUN);
10304 : 5570 DOCLN;
10305 : 5571 end;
10306 : 5572
10307 : 5573 ENDTST;
```

```
10315 035056 004167 146762 $T14: JSR R1,$SAVE2 ; 5451
10316 035062 005746 TST --(SP) ;
10317 035064 104402 1S: TRAP 2 ; 5470
10318 035066 152777 000040 156120 BISB #40,@ML.REG+40 ; 5472
10319 035074 016701 156502 MOV ML,DUT,R1 ;
10320 035100 042701 177770 BIC #177770,R1 ;
10321 035104 142777 000007 156102 BICB #7,@ML.REG+40 ;
10322 035112 150177 156076 BISB R1,@ML.REG+40 ;
10323 035116 005002 CLR R2 ; DODU.FLG 5474
10324 035120 004767 160714 JSR PC,FIRST.BLK.XFER ; 5475
10325 035124 012777 000061 156022 MOV #61,@ML.REG ; 5476
10326 035132 132777 000001 156014 BITB #1,@ML.REG ; 5478
10327 035140 001057 BNE 2S ;
10328 035142 104455 TRAP 55 ; 5481
10329 035144 000030 .WORD 30 ;
10330 035146 010464 .WORD ASYNC ;
10331 035150 024052 .WORD DUMPER ;
10332 035152 012746 006506 MOV #WRD.19,-(SP) ; 5482
10333 035156 012746 007774 MOV #FNC.5,-(SP) ;
10334 035162 012746 006426 MOV #WRD.12,-(SP) ;
10335 035166 012746 007366 MOV #PHR.1,-(SP) ;
10336 035172 012746 006276 MOV #WRD.1,-(SP) ;
10337 035176 012746 006062 MOV #FIV.FMT,-(SP) ;
10338 035202 012746 000006 MOV #6,-(SP) ;
10339 035206 010600 MOV SP,R0 ; SP,*
10340 035210 104414 TRAP 14 ;
10341 035212 132777 000200 156004 BITB #200,@ML.REG+50 ; 5484
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

Address	OpCode	Operand 1	Operand 2	Label	Comment	Seq
10343						
10344						
10345						
10346	035220	001056				
10347	035222	104455				
10348	035224	000031				
10349	035226	010464				
10350	035230	024052				
10351	035232	012746	007504			
10352	035236	012746	006276			
10353	035242	012746	006754			
10354	035246	012746	007366			
10355	035252	012746	006302			
10356	035256	012746	006062			
10357	035262	012746	000006			
10358	035266	010600				
10359	035270	104414				
10360	035272	062706	000016			
10361	035276	000427				
10362	035300	105777	155720	2S:		
10363	035304	100030				
10364	035306	104455				
10365	035310	000032				
10366	035312	010464				
10367	035314	024052				
10368	035316	012746	007472			
10369	035322	012746	006276			
10370	035326	012746	006754			
10371	035332	012746	007404			
10372	035336	012746	006302			
10373	035342	012746	006062			
10374	035346	012746	000006			
10375	035352	010600				
10376	035354	104414				
10377	035356	012702	000001	3S:		
10378	035362	062706	000016			
10379	035366	132777	000001	155640 4S:		
10380	035374	001430				
10381	035376	104455				
10382	035400	000033				
10383	035402	010464				
10384	035404	024052				
10385	035406	012746	006506			
10386	035412	012746	007774			
10387	035416	012746	006426			
10388	035422	012746	007472			
10389	035426	012746	006314			
10390	035432	012746	006062			
10391	035436	012746	000006			
10392	035442	010600				
10393	035444	104414				
10394	035446	012702	000001			
10395	035452	062706	000016			
10396	035456	032777	020000	155550 5S:		
10397	035464	001430				

```

10399                                     :ML4
10400                                     :
10401                                     :
10402 035466 104455                       TRAP      55
10403 035470 000034                       .WORD    34
10404 035472 010464                       .WORD    ASYNC
10405 035474 024052                       .WORD    DUMPER
10406 035476 012746 006506               MOV      #WORD.19,-(SP)
10407 035502 012746 007774               MOV      #FNC.5,-(SP)
10408 035506 012746 006426               MOV      #WORD.12,-(SP)
10409 035512 012746 007472               MOV      #PHR.5,-(SP)
10410 035516 012746 006322               MOV      #WORD.4,-(SP)
10411 035522 012746 006062               MOV      #FIV.FMT,-(SP)
10412 035526 012746 000006               MOV      #6,-(SP)
10413 035532 010600                       MOV      SP,R0
10414 035534 104414                       TRAP     14
10415 035536 012702 000001               MOV      #1,R2
10416 035542 062706 000016               ADD      #16,SP
10417 035546 012700 000050               6$: MOV   #50,R0
10418 035552 001410                       7$: BEQ   10$
10419 035554 016701 144336               MOV      LSDLY,R1
10420 035560 001401                       BEQ     9$
10421 035562 005016                       8$: CLR   (SP)
10422 035564 005301                       DEC     R1
10423 035566 001375                       BNE    8$
10424 035570 005300                       9$: DEC   R0
10425 035572 000767                       BR     7$
10426 035574 132777 000200 155422 10$: BITB  #200,@ML.REG+50
10427 035602 001106                       BNE    15$
10428 035604 132777 000001 155342       BITB  #1,@ML.REG
10429 035612 001452                       BEQ   13$
10430 035614 152777 000040 155372       BISB  #40,@ML.REG+40
10431 035622 016701 155754               MOV   ML,DUT,R1
10432 035626 042701 177770               BIC   #177770,R1
10433 035632 142777 000007 155354       BICB  #7,@ML.REG+40
10434 035640 150177 155350               BISB  R1,@ML.REG+40
10435 035644 132777 000001 155302       BITB  #1,@ML.REG
10436 035652 001405                       BEQ   11$
10437 035654 104455                       TRAP   55
10438 035656 000035                       .WORD  35
10439 035660 010464                       .WORD  ASYNC
10440 035662 024052                       .WORD  DUMPER
10441 035664 000404                       BR     12$
10442 035666 104455                       11$: TRAP  55
10443 035670 000036                       .WORD  36
10444 035672 010526                       .WORD  SYNC
10445 035674 024052                       .WORD  DUMPER
10446 035676 012746 006506               12$: MOV  #WORD.19,-(SP)
10447 035702 012746 007774               MOV  #FNC.5,-(SP)
10448 035706 012746 006416               MOV  #WORD.11,-(SP)
10449 035712 012746 007404               MOV  #PHR.2,-(SP)
10450 035716 012746 006276               MOV  #WORD.1,-(SP)
10451 035722 012746 006062               MOV  #FIV.FMT,-(SP)
10452 035726 012746 000006               MOV  #6,-(SP)
10453 035732 010600                       MOV  SP,R0

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

5514

5515

5516

5513

5519

5521

5525

5527

5530

5533

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

10455      :ML4
10456      :
10457
10458 035734 104414      TRAP      14
10459 035736 000424      BR        14$
10460 035740 104455      13$:     TRAP      55
10461 035742 000037      .WORD    37
10462 035744 010464      .WORD    ASYNC
10463 035746 024052      .WORD    DUMPER
10464 035750 012746 006506      MOV      #WORD.19,-(SP)
10465 035754 012746 007774      MOV      #FNC.5,-(SP)
10466 035756 012746 006416      MOV      #WORD.11,-(SP)
10467 035764 012746 007366      MOV      #PHR.1,-(SP)
10468 035770 012746 006302      MOV      #WORD.2,-(SP)
10469 035774 012746 006062      MOV      #FIV.FMT,-(SP)
10470 036000 012746 000006      MOV      #6,-(SP)
10471 036004 010600      MOV      SP,R0
10472 036006 104414      TRAP      14
10473 036010 012702 000001      14$:     MOV      #1,R2
10474 036014 062706 000016      ADD      #16,SP
10475 036020 132777 000001 155126 15$:     BITB     #1,@ML.REG
10476 036026 001455      BEQ      18$
10477 036030 152777 000040 155156      BISB     #40,@ML.REG+40
10478 036036 016701 155540      MOV      ML,DUT,R1
10479 036042 042701 177770      BIC      #177770,R1
10480 036046 142777 000007 155140      BICB     #7,@ML.REG+40
10481 036054 150177 155134      BISB     R1,@ML.REG+40
10482 036060 132777 000001 155066      BITB     #1,@ML.REG
10483 036066 001405      BEQ      16$
10484 036070 104455      TRAP      55
10485 036072 000040      .WORD    40
10486 036074 010464      .WORD    ASYNC
10487 036076 024052      .WORD    DUMPER
10488 036100 000404      BR        17$
10489 036102 104455      16$:     TRAP      55
10490 036104 000041      .WORD    41
10491 036106 010526      .WORD    SYNC
10492 036110 024052      .WORD    DUMPER
10493 036112 012746 006506      17$:     MOV      #WORD.19,-(SP)
10494 036116 012746 007774      MOV      #FNC.5,-(SP)
10495 036122 012746 006416      MOV      #WORD.11,-(SP)
10496 036126 012746 007404      MOV      #PHR.2,-(SP)
10497 036132 012746 006276      MOV      #WORD.1,-(SP)
10498 036136 012746 006062      MOV      #FIV.FMT,-(SP)
10499 036142 012746 000006      MOV      #6,-(SP)
10500 036146 010600      MOV      SP,R0
10501 036150 104414      TRAP      14
10502 036152 012702 000001      MOV      #1,R2
10503 036156 062706 000016      ADD      #16,SP
10504 036162 104467      18$:     TRAP      67
10505 036164 006000      ROR      R0
10506 036166 103002      BHIS     19$
10507 036170 000167 176670      JMP      1$
10508 036174 032777 040000 154752 19$:     BIT      #40000,@ML.REG
10509 036202 001432      BEQ      20$

```

5525
5537

5538

5541
5523
5544

5546

5549

5552

5553
5546
5554

5558

```

10511      :ML4
10512      :
10513      :
10514 036204 104455      TRAP      55      :
10515 036206 000163      .WORD    163      :
10516 036210 010672      .WORD    INTER    :
10517 036212 024052      .WORD    DUMPER   :
10518 036214 012746 006506      MOV      #WORD.19,-(SP) :
10519 036220 012746 007774      MOV      #FNC.5,-(SP)  :
10520 036224 012746 006426      MOV      #WORD.12,-(SP) :
10521 036230 012746 007472      MOV      #PHR.5,-(SP)  :
10522 036234 012746 006514      MOV      #WORD.20,-(SP) :
10523 036240 012746 007174      MOV      #WORD.61,-(SP) :
10524 036244 012746 006100      MOV      #SIX.FMT,-(SP) :
10525 036250 012746 000007      MOV      #7,-(SP)     :
10526 036254 010600      MOV      SP,R0        : SP,*
10527 036256 104414      TRAP     14          :
10528 036260 012702 000001      MOV      #1,R2        : *,DODU.FLG
10529 036264 062706 000020      ADD      #20,SP       :
10530 036270 005302      20$: DEC      R2        : DODU.FLG
10531 036272 001004      BNE      21$         :
10532 036274 016700 155300      MOV      ML.LUN,R0   :
10533 036300 104451      TRAP     51          :
10534 036302 104444      TRAP     44          :
10535 036304 005726      21$: TST      (SP)+     :
10536 036306 000207      RTS      PC          :
10537
10538      : Routine Size: 333 words
10539      : Maximum stack depth per invocation: 18 words
10544
10545
10549
10553 036310      T14::
10554 036310 004767 176542      1$: JSR      PC,$T14  :
10555 036314 104466      TRAP     66          :
10556 036316 006000      ROR      R0          :
10557 036320 103773      BLO     1$          :
10558 036322 000207      RTS      PC          :
10559
10560      : Routine Size: 6 words
10561      : Maximum stack depth per invocation: 0 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

5561

5562

5563

5560

5566

5569

5451

5571

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 253⁶³

SEQ 0240

10570
10571
10572 : 5574 !<BLF/PAGE>

10574 :ML4

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (65)

```

10575 :
10576 :
10577 :      5575 :
10578 :      5576 :
10579 :      5577 : BGNTST;
10580 :      5578 :
10581 :      5579 :
10582 :      5580 : TEST NUMBER: TST 15
10583 :      5581 :
10584 :      5582 : TEST NAME: READ FUNCTION TEST
10585 :      5583 :
10586 :      5584 : TEST DESCRIPTION:
10587 :      5585 :
10588 :      5586 : TEST IF THE DRIVE CAN PERFORM
10589 :      5587 : A READ FUNCTION WITHOUT
10590 :      5588 : HANGING THE DRIVE.
10591 :      5589 :
10592 :      5590 : A READ FUNCTION IS WRITTE*
10593 :      5591 : INTO MLCS1
10594 :      5592 :
10595 :      5593 : THEN GO AND ERROR BITS ARE
10596 :      5594 : CHECKED FOR CORRECT STATES.
10597 :      5595 :
10598 :      5596 :
10599 :      5597 :
10600 :      5598 : local
10601 :      5599 :     DODU_FLG;
10602 :      5600 :
10603 :      5601 : BGNSUB;
10604 :      5602 : CLR MBUS;
10605 :      5603 : DODU_FLG = ZERO;
10606 :      5604 : FIRST_BLK_XFER ();
10607 :      5605 : MLCS1 = read;
10608 :      5606 :
10609 :      5607 : if .GO IS_NOT_SET
10610 :      5608 : then
10611 :      5609 :     begin
10612 :      5610 :     ERRDF (34, ASYNC, DUMPER);
10613 :      5611 :     PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_6, WRD_19);
10614 :      5612 :
10615 :      5613 :     if .DRY IS_NOT_SET
10616 :      5614 :     then
10617 :      5615 :     begin
10618 :      5616 :     ERRDF (35, ASYNC, DUMPER);
10619 :      5617 :     PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
10620 :      5618 :     end;
10621 :      5619 :
10622 :      5620 :     DODU_FLG = ONE;
10623 :      5621 :     end
10624 :      5622 : else
10625 :      5623 :
10626 :      5624 :     if .DRY IS_SET
10627 :      5625 :     then
10628 :      5626 :     begin

```

!DROP UNIT FLAG

!SET UP A FIRST BLK XFERR
!DO A READ FUNCTION

!SEE IF GO GOT SET

!ERROR IF CLEAR

!TST DRY SET WITH GO CLEAR

!ERROR IF NOT SET

!SET DODU_FLG

!GO BIT GOT SET

!SEE IF DRY IS CLEAR

!ERROR IF SET

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BLJML4.BLI.2 (65)

```

10630 :ML4
10631 :
10632 :
10633 : 5627 ERRDF (36, ASYNC, DUMPER);
10634 : 5628 PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
10635 : 5629 DODU_FLG = ONE;
10636 : 5630 end;
10637 : 5631
10638 : 5632 if .ILF IS_SET !DID FUNCTION CAUSE ILF
10639 : 5633 then
10640 : 5634 begin !ERROR IF YES
10641 : 5635 ERRDF (37, ASYNC, DUMPER);
10642 : 5636 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_5, WRD_19);
10643 : 5637 DODU_FLG = ONE;
10644 : 5638 end;
10645 : 5639
10646 : 5640 if .OPI IS_SET !DID FUNCTION CAUSE OPI
10647 : 5641 then
10648 : 5642 begin !ERROR IF YES
10649 : 5643 ERRDF (38, ASYNC, DUMPER);
10650 : 5644 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_5, WRD_19);
10651 : 5645 DODU_FLG = ONE;
10652 : 5646 end;
10653 : 5647
10654 : 5648 DELAY (FRTY_US); !WAIT FOR XFERR TO COMPLETE
10655 : 5649
10656 : 5650 if .DRY IS_NOT_SET !IS DRY SET AFTER XFERR
10657 : 5651 then
10658 : 5652 begin
10659 : 5653
10660 : 5654 if .GO IS_SET !TEST GO CLEAR WITH DRY NOT SET
10661 : 5655 then
10662 : 5656 begin
10663 : 5657 CLR_MBUS; !CLEAR GO
10664 : 5658
10665 : 5659 if .GO IS_SET then ERRDF (39, ASYNC, DUMPER) else ERRDF (40, SYNC, DUMPER);
10666 : 5660
10667 : 5661 !TST GO TO DETERMINE FAILING MOD
10668 : 5662 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
10669 : 5663 end
10670 : 5664 else !GO BIT CLEAR WITH DRY NOT SET
10671 : 5665 begin !REPORT ERROR
10672 : 5666 ERRDF (41, ASYNC, DUMPER);
10673 : 5667 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_5, WRD_19);
10674 : 5668 end;
10675 : 5669
10676 : 5670 DODU_FLG = ONE; !SET DODU_FLG
10677 : 5671 end;
10678 : 5672
10679 : 5673 if .GO IS_SET !SEE IF GO CLEAR AFTER XFERR
10680 : 5674 then
10681 : 5675 begin
10682 : 5676 CLR_MBUS; !CLEAR GO IF STILL SET
10683 : 5677
10684 : 5678 if .GO IS_SET then ERRDF (42, ASYNC, DUMPER) else ERRDF (43, SYNC, DUMPER);

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (65)

```

10686 :ML4
10687 :
10688 :
10689 :          5679
10690 :          5680
10691 :          5681          PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
10692 :          5682          DODU_FLG = ONE;
10693 :          5683          end;
10694 :          5684
10695 :          5685          ENDSUB;
10696 :          5686
10697 :          5687          if .TRE IS_SET          !SEE IF XFERR CAUSED A TRANSFER ERROR
10698 :          5688          then
10699 :          5689          begin          !REPORT ERROR IF SET AND CONTINUE TESTING
10700 :          5690          ERRDF (116, INTER, DUMPER);
10701 :          5691          PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_6, WRD_19);
10702 :          5692          DODU_FLG = ONE;
10703 :          5693          end;
10704 :          5694
10705 :          5695          if .DODU_FLG IS_SET          !DROP THIS UNIT IF DODU_FLG SET
10706 :          5696          then
10707 :          5697          begin
10708 :          5698          DODU (.ML_LUN);
10709 :          5699          DOCLN;
10710 :          5700          end;
10711 :          5701
10712 :          5702          ENDTST;
10716 :
10720 036324 004167 145514          $T15: JSR R1, $SAVE2
10721 036330 005746          TST -(SP)
10722 036332 104402          1$: TRAP 2
10723 036334 152777 000040 154652          BISB #40, @ML.REG+40
10724 036342 016701 155234          MOV ML.DUT, R1
10725 036346 042701 177770          BIC #177770, R1
10726 036352 142777 000007 154634          BICB #7, @ML.REG+40
10727 036360 150177 154630          BISB R1, @ML.REG+40
10728 036364 005002          CLR R2
10729 036366 004767 157446          JSR PC, FIRST.BLK.XFER
10730 036372 012777 000071 154554          MOV #71, @ML.REG
10731 036400 132777 000001 154546          BITB #1, @ML.REG
10732 036406 001057          BNE 2$
10733 036410 104455          TRAP 55
10734 036412 000042          .WORD 42
10735 036414 010464          .WORD ASYNC
10736 036416 024052          .WORD DUMPER
10737 036420 012746 006506          MOV #WRD.19, -(SP)
10738 036424 012746 010004          MOV #FNC.6, -(SP)
10739 036430 012746 006426          MOV #WRD.12, -(SP)

```

5573
5599
5601
5603
5604
5605
5607
5610
5611

Address	Hex	Dec	Hex	Dec	Label	Code	Comment	Seq
10741								
10742								
10743								
10744	036434	012746	007366			MOV	#PHR.1,-(SP)	
10745	036440	012746	006276			MOV	#WRD.1,-(SP)	
10746	036444	012746	006062			MOV	#FV.FMT,-(SP)	
10747	036450	012746	000006			MOV	#6,-(SP)	
10748	036454	010600				MOV	SP,R0	: SP,*
10749	036456	104414				TRAP	14	
10750	036460	132777	000200	154536		BITB	#200,@ML.REG+50	: 5613
10751	036466	001056				BNE	3\$	
10752	036470	104455				TRAP	55	: 5616
10753	036472	000043				.WORD	43	
10754	036474	010464				.WORD	ASYN	
10755	036476	024052				.WORD	DUMPER	
10756	036500	012746	007504			MOV	#PHR.6,-(SP)	: 5617
10757	036504	012746	006276			MOV	#WRD.1,-(SP)	
10758	036510	012746	006754			MOV	#WRD.43,-(SP)	
10759	036514	012746	007366			MOV	#PHR.1,-(SP)	
10760	036520	012746	006302			MOV	#WRD.2,-(SP)	
10761	036524	012746	006062			MOV	#FV.FMT,-(SP)	
10762	036530	012746	000006			MOV	#6,-(SP)	
10763	036534	010600				MOV	SP,R0	: SP,*
10764	036536	104414				TRAP	14	
10765	036540	062706	000016			ADD	#16,SP	: 5615
10766	036544	000427				BR	3\$: 5620
10767	036546	105777	154452		2\$:	TSTB	@ML.REG+50	: 5624
10768	036552	100030				BPL	4\$	
10769	036554	104455				TRAP	55	: 5627
10770	036556	000044				.WORD	44	
10771	036560	010464				.WORD	ASYN	
10772	036562	024052				.WORD	DUMPER	
10773	036564	012746	007472			MOV	#PHR.5,-(SP)	: 5628
10774	036570	012746	006276			MOV	#WRD.1,-(SP)	
10775	036574	012746	006754			MOV	#WRD.43,-(SP)	
10776	036600	012746	007404			MOV	#PHR.2,-(SP)	
10777	036604	012746	006302			MOV	#WRD.2,-(SP)	
10778	036610	012746	006062			MOV	#FV.FMT,-(SP)	
10779	036614	012746	000006			MOV	#6,-(SP)	
10780	036620	010600				MOV	SP,R0	: SP,*
10781	036622	104414				TRAP	14	
10782	036624	012702	000001		3\$:	MOV	#1,R2	: *,DODU.FLG 5629
10783	036630	062706	000016			ADD	#16,SP	: 5626
10784	036634	132777	000001	154372	4\$:	BITB	#1,@ML.REG+60	: 5632
10785	036642	001430				BEQ	5\$	
10786	036644	104455				TRAP	55	: 5635
10787	036646	000045				.WORD	45	
10788	036650	010464				.WORD	ASYN	
10789	036652	024052				.WORD	DUMPER	
10790	036654	012746	006506			MOV	#WRD.19,-(SP)	: 5636
10791	036660	012746	007774			MOV	#FNC.5,-(SP)	
10792	036664	012746	006426			MOV	#WRD.12,-(SP)	
10793	036670	012746	007472			MOV	#PHR.5,-(SP)	
10794	036674	012746	006314			MOV	#WRD.3,-(SP)	
10795	036700	012746	006062			MOV	#FV.FMT,-(SP)	

22-Dec-1980 09:24:31 TOPS
 22-Dec-1980 09:21:22 PA:C

Address	Hex	Hex	Hex	Label	Code	Comment	Line
10797							
10798							
10799							
10800	036704	012746	000006		MOV	#6,-(SP)	
10801	036710	010600			MOV	SP,R0	: SP,*
10802	036712	104414			TRAP	14	
10803	036714	012702	000001		MOV	#1,R2	: *,DODU.FLG
10804	036720	062706	000016		ADD	#16,SP	:
10805	036724	032777	020000	154302 5\$:	BIT	#20000,@ML.REG+60	:
10806	036732	001430			BEQ	6\$:
10807	036734	104455			TRAP	55	: 5643
10808	036736	000046			.WORD	46	
10809	036740	010464			.WORD	ASYN	
10810	036742	024052			.WORD	DUMPER	
10811	036744	012746	006506		MOV	#WRD.19,-(SP)	: 5644
10812	036750	012746	007774		MOV	#FNC.5,-(SP)	
10813	036754	012746	006426		MOV	#WRD.12,-(SP)	
10814	036760	012746	007472		MOV	#PHR.5,-(SP)	
10815	036764	012746	006322		MOV	#WRD.4,-(SP)	
10816	036770	012746	006062		MOV	#FIV.FMT,-(SP)	
10817	036774	012746	000006		MOV	#6,-(SP)	
10818	037000	010600			MOV	SP,R0	: SP,*
10819	037002	104414			TRAP	14	
10820	037004	012702	000001		MOV	#1,R2	: *,DODU.FLG
10821	037010	062706	000016		ADD	#16,SP	:
10822	037014	012700	000050		MOV	#50,R0	: *,SSTMP2
10823	037020	001410		6\$:	BEQ	10\$:
10824	037022	016701	143070	7\$:	MOV	LSDLY,R1	: *,SSTMP1
10825	037026	001403			BEQ	9\$	
10826	037030	005016		8\$:	CLR	(SP)	: SSTMP
10827	037032	005301			DEC	R1	: SSTMP1
10828	037034	001375			BNE	8\$	
10829	037036	005300		9\$:	DEC	R0	: SSTMP2
10830	037040	000767			BR	7\$	
10831	037042	132777	000200	154154 10\$:	BITB	#200,@ML.REG+50	: 5650
10832	037050	001106			BNE	15\$	
10833	037052	132777	000001	154074	BITB	#1,@ML.REG	: 5654
10834	037060	001452			BEQ	13\$	
10835	037062	152777	000040	154124	BISB	#40,@ML.REG+40	: 5656
10836	037070	016701	154506		MOV	ML,DUT,R1	
10837	037074	042701	177770		BIC	#177770,R1	
10838	037100	142777	000007	154106	BICB	#7,@ML.REG+40	
10839	037106	150177	154102		BISB	R1,@ML.REG+40	
10840	037112	132777	000001	154034	BITB	#1,@ML.REG	: 5659
10841	037120	001405			BEQ	11\$	
10842	037122	104455			TRAP	55	
10843	037124	000047			.WORD	47	
10844	037126	010464			.WORD	ASYN	
10845	037130	024052			.WORD	DUMPER	
10846	037132	000404			BR	12\$	
10847	037134	104455		11\$:	TRAP	55	
10848	037136	000050			.WORD	50	
10849	037140	010526			.WORD	SYN	
10850	037142	024052			.WORD	DUMPER	
10851	037144	012746	006506	12\$:	MOV	#WRD.19,-(SP)	: 5662

Address	Hex	Hex	Hex	Label	Instruction	Comments	Line
10853							
10854							
10855							
10856	037150	012746	007774		MOV #FNC.5,-(SP)		
10857	037154	012746	006416		MOV #WORD.11,-(SP)		
10858	037160	012746	007404		MOV #PHR.2,-(SP)		
10859	037164	012746	006276		MOV #WORD.1,-(SP)		
10860	037170	012746	006062		MOV #FIV.FMT,-(SP)		
10861	037174	012746	000006		MOV #6,-(SP)		
10862	037200	010600			MOV SP,R0	: SP,*	
10863	037202	104414			TRAP 14	:	
10864	037204	00C424			BR 14\$:	5654
10865	037206	104455		13\$:	TRAP 55	:	5666
10866	037210	000051			.WORD 51	:	
10867	037212	010464			.WORD ASYNC	:	
10868	037214	024052			.WORD DUMPER	:	
10869	037216	012746	006506		MOV #WORD.19,-(SP)	:	5667
10870	037222	012746	007774		MOV #FNC.5,-(SP)	:	
10871	037226	012746	006416		MOV #WORD.11,-(SP)	:	
10872	037232	012746	007366		MOV #PHR.1,-(SP)	:	
10873	037236	012746	006302		MOV #WORD.2,-(SP)	:	
10874	037242	012746	006062		MOV #FIV.FMT,-(SP)	:	
10875	037246	012746	000006		MOV #6,-(SP)	:	
10876	037252	010600			MOV SP,R0	: SP,*	
10877	037254	104414			TRAP 14	:	
10878	037256	012702	000001	14\$:	MOV #1,R2	: *,DODU.FLG	5670
10879	037262	062706	000016		ADD #16,SP	:	5652
10880	037266	132777	000001	153660 15\$:	BITB #1,@ML.REG	:	5673
10881	037274	001455			BEQ 18\$:	
10882	037276	152777	000040	153710	BISB #40,@ML.REG+40	:	5675
10883	037304	016701	154272		MOV ML.DUT,R1	:	
10884	037310	042701	177770		BIC #177770,R1	:	
10885	037314	142777	000007	153672	BICB #7,@ML.REG+40	:	
10886	037322	150177	153666		BISB R1,@ML.REG+40	:	
10887	037326	132777	000001	153620	BITB #1,@ML.REG	:	5678
10888	037334	001405			BEQ 16\$:	
10889	037336	104455			TRAP 55	:	
10890	037340	030052			.WORD 52	:	
10891	037342	010464			.WORD ASYNC	:	
10892	037344	024052			.WORD DUMPER	:	
10893	037346	000404			BR 17\$:	
10894	037350	104455		16\$:	TRAP 55	:	
10895	037352	000053			.WORD 53	:	
10896	037354	010526			.WORD SYNC	:	
10897	037356	024052			.WORD DUMPER	:	
10898	037360	012746	006506	17\$:	MOV #WORD.19,-(SP)	:	5681
10899	037364	012746	007774		MOV #FNC.5,-(SP)	:	
10900	037370	012746	006416		MOV #WORD.11,-(SP)	:	
10901	037374	012746	007404		MOV #PHR.2,-(SP)	:	
10902	037400	012746	006276		MOV #WORD.1,-(SP)	:	
10903	037404	012746	006062		MOV #FIV.FMT,-(SP)	:	
10904	037410	012746	000006		MOV #6,-(SP)	:	
10905	037414	010600			MOV SP,R0	: SP,*	
10906	037416	104414			TRAP 14	:	
10907	037420	012702	000001		MOV #1,R2	: *,DODU.FLG	5682

```

10909      ;ML4
10910      ;
10911
10912 037424 062706 000016      ADD      #16,SP      ;
10913 037430 104467      18$: TRAP      67      ;
10914 037432 006000      ROR      R0      ;
10915 037434 103002      BHIS     19$
10916 037436 000167 176670      JMP      1$
10917 037442 032777 040000 153504 19$: BIT      #40000,@ML.REG ;
10918 037450 001432      BEQ      20$
10919 037452 104455      TRAP     55      ;
10920 037454 000164      .WORD   164
10921 037456 010672      .WORD   INTER
10922 037460 024052      .WORD   DUMPER
10923 037462 012746 006506      MOV      #WORD.19,-(SP) ;
10924 037466 012746 010004      MOV      #FNC.6,-(SP)
10925 037472 012746 006426      MOV      #WORD.12,-(SP)
10926 037476 012746 007472      MOV      #PHR.5,-(SP)
10927 037502 012746 006514      MOV      #WORD.20,-(SP)
10928 037506 012746 007174      MOV      #WORD.61,-(SP)
10929 037512 012746 006100      MOV      #SIX.FMT,-(SP)
10930 037516 012746 000007      MOV      #7,-(SP)
10931 037522 010600      MOV      SP,R0      ; SP,*
10932 037524 104414      TRAP     14
10933 037526 012702 000001      MOV      #1,R2      ; *,DODU.FLG
10934 037532 062706 000020      ADD      #20,SP      ;
10935 037536 005302      20$: DEC      R2      ; DODU.FLG
10936 037540 001004      BNE
10937 037542 016700 154032      MOV      ML.LUN,R0 ;
10938 037546 104451      TRAP     51
10939 037550 104444      TRAP     44
10940 037552 005726      21$: TST      (SP)+ ;
10941 037554 000207      RTS      PC
10942
10943      ; Routine Size: 333 words
10944      ; Maximum stack depth per invocation: 18 words
10949
10950
10954
10958 037556      T15::
10959 037556 004767 176542      1$: JSR      PC,$T15 ;
10960 037562 104466      TRAP     66
10961 037564 006000      ROR      R0
10962 037566 103773      BLO      1$

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

5675
5683

5687

5690

5691

5692

5689

5695

5698

5573

5700

10964
10965
10966
10967 037570 000207
10968
10969
10970
10975
10976
10977 ;

:ML4
:
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

5703 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (66)

```

10979 :ML4
10980 :
10981 :
10982 : 5704 !
10983 : 5705 !BGNTST;
10984 : 5706 !
10985 : 5707 !++
10986 : 5708 !TEST NUMBER: TST 16
10987 : 5709 !
10988 : 5710 !TEST NAME: CLEAR FUNCTION TEST
10989 : 5711 !
10990 : 5712 !TEST DESCRIPTION:
10991 : 5713 !TEST IF THE DRIVE CAN PERFORM A CLEAR FUNCTION WITHOUT HANGING THE DRIVE.
10992 : 5714 !
10993 : 5715 !A CLEAR FUNCTION IS WRITTEN INTO MLCS1.
10994 : 5716 !
10995 : 5717 !THEN GO AND ERROR BITS ARE CHECKED FOR CORRECT STATUS.
10996 : 5718 !THIS DRIVE IS DROPPED ON DETECTED ERRORS.
10997 : 5719 !
10998 : 5720 !--
10999 : 5721 !
11000 : 5722 !local
11001 : 5723 !DODU_FLG; !DROP UNIT FLAG
11002 : 5724 !
11003 : 5725 !BGNSUB;
11004 : 5726 !CLR MBUS;
11005 : 5727 !DODU_FLG = ZERO;
11006 : 5728 !MLER = ONES; !SET BITS IN ERROR REGISTER
11007 : 5729 !MLCS1 = DRV CLR; !DO A CLEAR FUNCTION
11008 : 5730 !DELAY (ONE_DS); !DELAY
11009 : 5731 !
11010 : 5732 !if .GO IS_SET !SEE IF GO CLEARED AFTER FUNCTION
11011 : 5733 !then
11012 : 5734 !begin !ERROR IF SET
11013 : 5735 !ERRDF (44, ASYNC, DUMPER);
11014 : 5736 !PRINTB (FIV_FMT, WRD_1, PHR_5, WRD_11, FNC_3, FNC_7, WRD_19);
11015 : 5737 !
11016 : 5738 !if .DRY IS_SET !TST DRY CLEAR WITH GO SET
11017 : 5739 !then
11018 : 5740 !begin !ERROR IF SET
11019 : 5741 !ERRDF (45, ASYNC, DUMPER);
11020 : 5742 !PPRINTB (FIV_FMT, WRD_2, PHR_5, WRD_43, WRD_1, PHR_5);
11021 : 5743 !end;
11022 : 5744 !
11023 : 5745 !DODU_FLG = ONE; !SET DODU_FLG
11024 : 5746 !end
11025 : 5747 !else !GO CLEARED AFTER FUNCTION
11026 : 5748 !
11027 : 5749 !if .DRY IS_NOT_SET !TST DRY SET WITH GO CLEAR
11028 : 5750 !then
11029 : 5751 !begin !ERROR IF NOT SET
11030 : 5752 !ERRDF (46, ASYNC, DUMPER);
11031 : 5753 !PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
11032 : 5754 !DODU_FLG = ONE;
11033 : 5755 !end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (66)

```

11035 :ML4
11036 :
11037 :
11038 : 5756
11039 : 5757 if .ILF IS_SET           !DID FUNCTION CAUSE ILF
11040 : 5758 then
11041 : 5759   begin                 !ERROR IF YES
11042 : 5760   ERRDF (47, ASYNC, DUMPER);
11043 : 5761   PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);
11044 : 5762   DODU_FLG = ONE;
11045 : 5763   end;
11046 : 5764
11047 : 5765 if .OPI IS_SET           !DID FUNCTION CAUSE OPI
11048 : 5766 then
11049 : 5767   begin                 !ERROR IF YES
11050 : 5768   ERRDF (48, ASYNC, DUMPER);
11051 : 5769   PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);
11052 : 5770   DODU_FLG = ONE;
11053 : 5771   end;
11054 : 5772
11055 : 5773 if .MLER neq ZERO       !TEST ERROR REGISTER FOR CLEAR
11056 : 5774 then
11057 : 5775   begin                 !ERROR IF NOT CLEAR
11058 : 5776   ERRDF (49, ASYNC, DUMPER);
11059 : 5777   PRINTB (SIX_FMT, FNC_3, FNC_7, WRD_19, WRD_14, WRD_13, REG_3);
11060 : 5778   end;
11061 : 5779
11062 : 5780 ENDSUB;
11063 : 5781
11064 : 5782 if .DODU_FLG IS_SET     !DROP THIS UNIT IF DODU_FLG SET
11065 : 5783 then
11066 : 5784   begin
11067 : 5785   DODU (.ML_LUN);
11068 : 5786   DOCLN;
11069 : 5787   end;
11070 : 5788
11071 : 5789 ENDTST;

```

```

11079 037572 004167 144246      $T16: JSR      R1,$SAVE2           :           5702
11080 037576 005746                TST      -(SP)                :
11081 037600 104402                TRAP    2                      :           5723
11082 037602 152777 000040 153404      BISB    #40,@ML.REG+40        :           5725
11083 037610 016701 153766                MOV     ML,DUT,R1             :
11084 037614 042701 177770                BIC     #177770,R1            :
11085 037620 142777 000007 153366      BICB    #7,@ML.REG+40         :
11086 037626 150177 153362                BISB    R1,@ML.REG+40         :
11087 037632 005002                CLR     R2                     : DODU.FLG  5727
11088 037634 012777 177777 153372      MOV     #-1,@ML.REG+60        :           5728

```

Line	Address	Offset	Value	Label	Op	Opnd	Comment	Seq
11090				:ML4				
11091				:				
11092				:				
11093	037642	012777	000011	153304	MOV	#11,@ML.REG	:	5729
11094	037650	012700	000001		MOV	#1,R0	: *,SSTMP2	5730
11095	037654	001410		2S:	BEQ	5S	:	
11096	037656	016701	142234		MOV	LSDLY,R1	: *,SSTMP1	
11097	037662	001403			BEQ	4S	:	
11098	037664	005016		3S:	CLR	(SP)	: SSTMP	
11099	037666	005301			DEC	R1	: SSTMP1	
11100	037670	001375			BNE	3S	:	
11101	037672	005300		4S:	DEC	R0	: SSTMP2	
11102	037674	000767			BR	2S	:	
11103	037676	132777	000001	153250	BITB	#1,@ML.REG	:	5732
11104	037704	001464			BEQ	7S	:	
11105	037706	104455			TRAP	55	:	5735
11106	037710	000054			.WORD	54	:	
11107	037712	010464			.WORD	ASYNC	:	
11108	037714	024052			.WORD	DUMPER	:	
11109	037716	012746	006506		MOV	#WORD.19,-(SP)	:	5736
11110	037722	012746	010012		MOV	#FNC.7,-(SP)	:	
11111	037726	012746	007750		MOV	#FNC.3,-(SP)	:	
11112	037732	012746	006416		MOV	#WORD.11,-(SP)	:	
11113	037736	012746	007472		MOV	#PHR.5,-(SP)	:	
11114	037742	012746	006276		MOV	#WORD.1,-(SP)	:	
11115	037746	012746	006062		MOV	#FIV.FMT,-(SP)	:	
11116	037752	012746	000007		MOV	#7,-(SP)	:	
11117	037756	010600			MOV	SP,R0	: SP,*	
11118	037760	104414			TRAP	14	:	
11119	037762	105777	153236		TSTB	@ML.REG+50	:	5738
11120	037766	100026			BPL	6S	:	
11121	037770	104455			TRAP	55	:	5741
11122	037772	000055			.WORD	55	:	
11123	037774	010464			.WORD	ASYNC	:	
11124	037776	024052			.WORD	DUMPER	:	
11125	040000	012746	007472		MOV	#PHR.5,-(SP)	:	5742
11126	040004	012746	006276		MOV	#WORD.1,-(SP)	:	
11127	040010	012746	006754		MOV	#WORD.43,-(SP)	:	
11128	040014	012746	007472		MOV	#PHR.5,-(SP)	:	
11129	040020	012746	006302		MOV	#WORD.2,-(SP)	:	
11130	040024	012746	006062		MOV	#FIV.FMT,-(SP)	:	
11131	040030	012746	000006		MOV	#6,-(SP)	:	
11132	040034	010600			MOV	SP,R0	: SP,*	
11133	040036	104414			TRAP	14	:	
11134	040040	062706	000016		ADD	#16,SP	:	5740
11135	040044	012702	000001	6S:	MOV	#1,R2	: *,DODU.FLG	5745
11136	040050	062706	000020		ADD	#20,SP	:	5734
11137	040054	000434			BR	8S	:	5732
11138	040056	132777	000200	153140	BITB	#200,@ML.REG+50	:	5749
11139	040064	001030			BNE	8S	:	
11140	040066	104455			TRAP	55	:	5752
11141	040070	000056			.WORD	56	:	
11142	040072	010464			.WORD	ASYNC	:	
11143	040074	024052			.WORD	DUMPER	:	
11144	040076	012746	007504		MOV	#PHR.6,-(SP)	:	5753

```

11146          :ML4
11147          :
11148          :
11149 040102 012746 006276      MOV      #WRD.1,-(SP)
11150 040106 012746 006754      MOV      #WRD.43,-(SP)
11151 040112 012746 007366      MOV      #PHR.1,-(SP)
11152 040116 012746 006302      MOV      #WRD.2,-(SP)
11153 040122 012746 006062      MOV      #FIV.FMT,-(SP)
11154 040126 012746 000006      MOV      #6,-(SP)
11155 040132 010600      MOV      SP,R0          ; SP,*
11156 040134 104414      TRAP     14
11157 040136 012702 000001      MOV      #1,R2          ; *,DODU.FLG
11158 040142 062706 000016      ADD      #15,SP          ;
11159 040146 132777 000001 153060 8$:  BITB     #1,@ML.REG+60  ;
11160 040154 001432      BEQ      98             ;
11161 040156 104455      TRAP     55             ;
11162 040160 000057      .WORD   57             ;
11163 040162 010464      .WORD   ASYNC           ;
11164 040164 024052      .WORD   DUMPER          ;
11165 040166 012746 006506      MOV      #WRD.19,-(SP)  ;
11166 040172 012746 010012      MOV      #FNC.7,-(SP)  ;
11167 040176 012746 007750      MOV      #FNC.3,-(SP)  ;
11168 040202 012746 006426      MOV      #WRD.12,-(SP) ;
11169 040206 012746 007472      MOV      #PHR.5,-(SP)  ;
11170 040212 012746 006314      MOV      #WRD.3,-(SP)  ;
11171 040216 012746 006062      MOV      #FIV.FMT,-(SP)
11172 040222 012746 000007      MOV      #7,-(SP)
11173 040226 010600      MOV      SP,R0          ; SP,*
11174 040230 104414      TRAP     14
11175 040232 012702 000001      MOV      #1,R2          ; *,DODU.FLG
11176 040236 062706 000020      ADD      #20,SP          ;
11177 040242 032777 020000 152764 9$:  BIT      #20000,@ML.REG+60 ;
11178 040250 001432      BEQ      108            ;
11179 040252 104455      TRAP     55             ;
11180 040254 000060      .WORD   60             ;
11181 040256 010464      .WORD   ASYNC           ;
11182 040260 024052      .WORD   DUMPER          ;
11183 040262 012746 006506      MOV      #WRD.19,-(SP)  ;
11184 040266 012746 010012      MOV      #FNC.7,-(SP)  ;
11185 040272 012746 007750      MOV      #FNC.3,-(SP)  ;
11186 040276 012746 006426      MOV      #WRD.12,-(SP) ;
11187 040302 012746 007472      MOV      #PHR.5,-(SP)  ;
11188 040306 012746 006322      MOV      #WRD.4,-(SP)  ;
11189 040312 012746 006062      MOV      #FIV.FMT,-(SP)
11190 040316 012746 000007      MOV      #7,-(SP)
11191 040322 010600      MOV      SP,R0          ; SP,*
11192 040324 104414      TRAP     14
11193 040326 012702 000001      MOV      #1,R2          ; *,DODU.FLG
11194 040332 062706 000020      ADD      #20,SP          ;
11195 040336 005777 152672 10$:  TST      @ML.REG+60     ;
11196 040342 001430      BEQ      118            ;
11197 040344 104455      TRAP     55             ;
11198 040346 000061      .WORD   61             ;
11199 040350 010464      .WORD   ASYNC           ;
11200 040352 024052      .WORD   DUMPER          ;

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

11202      ;ML4
11203      ;
11204      ;
11205 040354 012746 010274      MOV    #REG.3,-(SP)      ;
11206 040360 012746 006436      MOV    #WRD.13,-(SP)
11207 040364 012746 006442      MOV    #WRD.14,-(SP)
11208 040370 012746 006506      MOV    #WRD.19,-(SP)
11209 040374 012746 010012      MOV    #FNC.7,-(SP)
11210 040400 012746 007750      MOV    #FNC.3,-(SP)
11211 040404 012746 006100      MOV    #SIX.FMT,-(SP)
11212 040410 012746 000007      MOV    #7,-(SP)
11213 040414 010600      MOV    SP,R0      ; SP,*
11214 040416 104414      TRAP   14
11215 040420 062706 000020      ADD    #20,SP
11216 040424 104467      11$:  TRAP   67
11217 040426 006000      ROR    R0
11218 040430 103002      BHIS   12$
11219 040432 000167 177142      JMP    1$
11220 040436 005302      12$:  DEC    R2      ; DODU.FLG
11221 040440 001004      BNE    13$
11222 040442 016700 153132      MOV    ML.LUN,R0
11223 040446 104451      TRAP   51
11224 040450 104444      TRAP   44
11225 040452 005726      13$:  TST    (SP)+
11226 040454 000207      RTS    PC
11227
11228      ; Routine Size: 218 words
11229      ; Maximum stack depth per invocation: 19 words
11234
11235
11239
11243 040456      T16::
11244 040456 004767 177110      1$:  JSR    PC,$T16
11245 040462 104466      TRAP   66
11246 040464 006000      ROR    R0
11247 040466 103773      BLO    1$
11248 040470 000207      RTS    PC
11249
11250      ; Routine Size: 6 words
11251      ; Maximum stack depth per invocation: 0 words

```


CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 267^{H 4}

SEQ 0254

11257 ;ML4
11258 ;
11259 ;
11260 ;

5790 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (66)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (67)

```

11262 :ML4
11263 :
11264 :
11265 : 5791 !
11266 : 5792 !BGNTST
11267 : 5793 !
11268 : 5794 !++
11269 : 5795 ! TEST NUMBER: TST 17
11270 : 5796 !
11271 : 5797 ! TEST NAME: DIAGNOSTIC REGISTER TESTS
11272 : 5798 !
11273 : 5799 ! TEST DESCRIPTION:
11274 : 5800 ! TEST THE DATA DIAG REGISTERS MLD1, MLD2, MLE2
11275 : 5801 ! FOR 1'S/O'S, SHIFTING 1'S/O'S AND INITIALIZATION
11276 : 5802 !
11277 : 5803 !
11278 : 5804 !
11279 : 5805 ! local
11280 : 5806 ! CLR_DATA , !CLEAR DATA FOR INIT TEST
11281 : 5807 ! SAVE !TEMPORARY SAVE LOCATION
11282 : 5808 ! ERR_FLG !ERROR FLAG
11283 : 5809 ! TST_PAT !TEST PATTERN
11284 : 5810 ! index !POINTS TO REGISTER PRESENTLY BEING TESTED
11285 : 5811 ! DODU_FLG !DROP UNIT FLAG
11286 : 5812 ! DODU_FLG = ZERO;
11287 : 5813 !
11288 : 5814 !
11289 : 5815 ! FIRST TEST THE REGISTERS FOR ONES AND ZEROES
11290 : 5816 !
11291 : 5817 !
11292 : 5818 ! TST_PAT = ONES; !LOAD TEST PAT WITH ONES
11293 : 5819 !
11294 : 5820 ! incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
11295 : 5821 ! begin
11296 : 5822 !
11297 : 5823 ! incr REG_SEL from 11 to 13 do !TEST ELEVEN WRITE/READ REGISTERS
11298 : 5824 ! begin
11299 : 5825 ! BGNSUB;
11300 : 5826 ! CLR_MBUS;
11301 : 5827 ! WRT_REG (.TST_PAT, .REG_SEL, index); !WRITE TO THE REGISTER
11302 : 5828 ! RD_REG (.TST_PAT, .REG_SEL, ERR_FLG); !READ THE REGISTER
11303 : 5829 !
11304 : 5830 ! if .ERR_FLG IS_SET !SEE IF READ FOUND AN ERROR
11305 : 5831 ! then
11306 : 5832 ! begin !IF ERROR FLAG IS SET THEN ERROR AND SET DODU_FLG
11307 : 5833 ! ERRDF (121, ARR_DAT, DUMPER); !ARRAY DATA MODULE FAILURE
11308 : 5834 ! PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
11309 : 5835 ! PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
11310 : 5836 ! DODU_FLG = ONE;
11311 : 5837 ! end;
11312 : 5838 !
11313 : 5839 ! ENDSUB;
11314 : 5840 ! end;
11315 : 5841 !
11316 : 5842 ! TST_PAT = not .TST_PAT; !REPEAT AGAIN WITH COMPLIMENT DATA

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (67)

```

11318 :ML4
11319 :
11320 :
11321 :      5843      end:
11322 :      5844
11323 :      5845      |
11324 :      5846      |
11325 :      5847      |      TEST THE REGISTERS FOR SHIFTING ONES AND ZEROES
11326 :      5848      |
11327 :      5849      |
11328 :      5850      TST_PAT = ONE;          !LOAD TST_PAT WITH A 1 IN A FIELD OF 0'S
11329 :      5851
11330 :      5852      incr SHIFT from 0 to 15 do          !DO SHIFT 16 TIMES
11331 :      5853      begin
11332 :      5854
11333 :      5855      incr TWICE from 0 to 1 do          !REPEAT LOOP TWICE
11334 :      5856      begin
11335 :      5857
11336 :      5858      incr REG_SEL from 11 to 13 do          !TEST ELEVEN READ/WRITE REGISTERS
11337 :      5859      begin
11338 :      5860      BGN SUB;
11339 :      5861      CLR_MBUS;
11340 :      5862      WRT_REG (.TST_PAT, .REG_SEL, index);          !WRITE TO THE REGISTER
11341 :      5863      RD_REG (.TST_PAT, .REG_SEL, ERR_FLG);          !READ THE REGISTER
11342 :      5864
11343 :      5865      if .ERR_FLG IS_SET          !SEE IF THE READ FOUND AN ERROR
11344 :      5866      then
11345 :      5867      begin
11346 :      5868      ERR_DF (122, ARR_DAT, DUMPER);          !IF THE ERROR FLAG IS SET THEN ERROR
11347 :      5869      PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);          !ARRAY DATA MODULE FAILURE
11348 :      5870      PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
11349 :      5871      DDU_FLG = ONE;
11350 :      5872      end;
11351 :      5873
11352 :      5874      ENDSUB;
11353 :      5875      end;
11354 :      5876
11355 :      5877      TST_PAT = not .TST_PAT;          !REPEAT WITH A 0 IN A FIELD OF 1'S
11356 :      5878      end;
11357 :      5879
11358 :      5880      TST_PAT = .TST_PAT^ONE;          !SHIFT THE 1 IN THE FIELD OF 0'S
11359 :      5881      end;
11360 :      5882
11361 :      5883      |
11362 :      5884      |
11363 :      5885      |      NOW TEST THE REGISTERS FOR INITIALIZATION
11364 :      5886      |
11365 :      5887      |
11366 :      5888      REG_INIT_FLG = ONE;
11367 :      5889      TST_PAT = ONES;          !BACKGROUND PATTERN
11368 :      5890
11369 :      5891      incr TWICE from 0 to 1 do          !REPEAT LOOP TWICE
11370 :      5892      begin
11371 :      5893
11372 :      5894      incr REG_SEL from 11 to 13 do          !TEST THIRTEEN REGISTERS

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (67)

```

11374 :ML4
11375 :
11376 :
11377 :      5895      begin
11378 :      5896      BGN SUB;
11379 :      5897      CLR MEUS;
11380 :      5898      WRT_REG (.TST_PAT, .REG_SEL, index);      !WRITE REGISTER WITH BACKGROUND
11381 :      5899      CLR_DATA = (.RI) or (.IGNORE);          !CALCULATE THE CLEARED DATA PATTERN
11382 :      5900      REG_INIT_FLG = ONE;
11383 :      5901      RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);    !READ THE REGISTER FOR THE CLEARED DATA PAT
11384 :      5902
11385 :      5903      if .ERR_FLG IS_SET                          !SEE IF READ FOUND AN ERROR
11386 :      5904      then
11387 :      5905      begin                                      !IF ERROR FLAG IS SET THEN ERROR AND SET DODU_FLG
11388 :      5906      ERRDF (118, ARR_DAT, DUMPER);              !ARRAY DATA MODULE
11389 :      5907      PRINTB (FIV_FMT, PHR 4, WRD 12, WRD 52, FNC 23, WRD 56);
11390 :      5908      PRINTB (FMT_16, .ML_REG [index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
11391 :      5909      DODU_FLG = ONE;
11392 :      5910      end;
11393 :      5911
11394 :      5912      ENDSUB;
11395 :      5913      end;
11396 :
11397 :      5915      TST_PAT = not .TST_PAT;                    !REPEAT WITH COMPLIMENT BACKGROUND PAT
11398 :      5916      end;
11399 :      5917
11400 :      5918      REG_INIT_FLG = ZERO;                        !CLEAR THE FLAG
11401 :      5919
11402 :      5920      if .DODU_FLG IS_SET                          !DROP THIS UNIT IF DODU_FLG SET
11403 :      5921      then
11404 :      5922      begin
11405 :      5923      DODU (.ML_LUN);
11406 :      5924      DOCLN;
11407 :      5925      end;
11408 :      5926
11409 :      5927      ENDTST;
11410 :
11411 :
11412 :
11413 :
11414 :
11415 :
11416 :
11417 : 040472 004167 143420      ST17: JSR      R1, $SAVE5      ;
11418 : 040476 024646              CMP      -(SP), -(SP)      ;
11419 : 040500 005046              CLR      -(SP)              ; DODU.FLG
11420 : 040502 012703 177777      MOV      #-1, R3            ; *,TST.PAT
11421 : 040506 005001              CLR      R1                  ; TWICE
11422 : 040510 012702 000013      1$: MOV      #13, R2        ; *,REG.SEL
11423 : 040514 104402              2$: TRAP     2              ;
11424 : 040516 152777 000040 152470 B1SB     #40, @ML.REG+40      ;
11425 : 040524 016705 153052      MOV      ML.DUT, R5         ;
11426 : 040530 042705 177770      BIC      #177770, R5        ;
11427 : 040534 142777 000007 152452 B1CB     #7, @ML.REG+40      ;

```

5789
5812
5818
5820
5823
5824
5825

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

Address	OpCode	Operand 1	Operand 2	Operand 3	Comment	Seq
11429						
11430						
11431						
11432	040542	150577	152446		BISB R5, @ML.REG+40	
11433	040546	010346			MOV R3, -(SP) ; TST.PAT,*	5827
11434	040550	010246			MOV R2, -(SP) ; REG.SEL,*	
11435	040552	012746	000012		MOV #12, -(SP)	
11436	040556	060616			ADD SP, (SP) ; INDEX,*	
11437	040560	004767	162310		JSR PC, WRT.REG	
11438	040564	010316			MOV R3, (SP) ; TST.PAT,*	5828
11439	040566	010246			MOV R2, -(SP) ; REG.SEL,*	
11440	040570	012746	000014		MOV #14, -(SP)	
11441	040574	060616			ADD SP, (SP) ; ERR.FLG,*	
11442	040576	004767	162660		JSR PC, RD.REG	
11443	040602	026627	000014	000001	CMP 14(SP), #1 ; ERR.FLG,*	5830
11444	040610	001054			BNE 3\$	
11445	040612	104455			TRAP 55 ;	5833
11446	040614	000171			.WORD 171	
11447	040616	010570			.WORD ARR.DAT	
11448	040620	024052			.WORD DUMPER	
11449	040622	012746	007122		MOV #WORD.56, -(SP) ;	5834
11450	040626	012746	007066		MOV #WORD.52, -(SP)	
11451	040632	012746	010004		MOV #FNC.6, -(SP)	
11452	040636	012746	007774		MOV #FNC.5, -(SP)	
11453	040642	012746	006426		MOV #WORD.12, -(SP)	
11454	040646	012746	007454		MOV #PHR.4, -(SP)	
11455	040652	012746	006100		MOV #SIX.FMT, -(SP)	
11456	040656	012746	000007		MOV #7, -(SP)	
11457	040662	010600			MOV SP, R0 ; SP,*	
11458	040664	104414			TRAP 14	
11459	040666	016716	152236		MOV RD.DATA, (SP) ;	5835
11460	040672	016746	152230		MOV WT.DATA, -(SP)	
11461	040676	016600	000040		MOV 40(SP), R0 ; INDEX,*	
11462	040702	006300			ASL R0	
11463	040704	006300			ASL R0	
11464	040706	006300			ASL R0	
11465	040710	016046	013154		MOV ML.REG(R0), -(SP)	
11466	040714	012746	005216		MOV #FMT.16, -(SP)	
11467	040720	012746	000004		MOV #4, -(SP)	
11468	040724	010600			MOV SP, R0 ; SP,*	
11469	040726	104414			TRAP 14	
11470	040730	012766	000001	000042	MOV #1, 42(SP) ; *,DODU.FLG	5836
11471	040736	062706	000030		ADD #30, SP ;	5832
11472	040742	062706	000012		ADD #12, SP ;	5824
11473	040746	104467			TRAP 67 ;	5837
11474	040750	006000			ROR R0	
11475	040752	103660			BLO 2\$	
11476	040754	005202			INC R2 ; REG.SEL	5823
11477	040756	020227	000015		CMP R2, #15 ; REG.SEL,*	
11478	040762	003654			BLE 2\$	
11479	040764	005103			COM R3 ; TST.PAT	5842
11480	040766	005201			INC R1 ; TWICE	5820
11481	040770	020127	000001		CMP R1, #1 ; TWICE,*	
11482	040774	003645			BLE 1\$	
11483	040776	012703	000001		MOV #1, R3 ; *,TST.PAT	5850

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Line No.
11485				:ML4			
11486				:			
11487				:			
11488	041002	005004			CLR R4	: SHIFT	5852
11489	041004	005001		48:	CLR R1	: TWICE	5855
11490	041006	012702	000013	58:	MOV #13,R2	: *,REG.SEL	5858
11491	041012	104402		68:	TRAP 2	:	5859
11492	041014	152777	000040	152172	BISB #40,@ML.REG+40	:	5860
11493	041022	016705	152554		MOV ML,DUT,R5		
11494	041026	042705	177770		BIC #177770,R5		
11495	041032	142777	000007	152154	BICB #7,@ML.REG+40		
11496	041040	150577	152150		BISB R5,@ML.REG+40		
11497	041044	010346			MOV R3,-(SP)	: TST.PAT,*	5862
11498	041046	010246			MOV R2,-(SP)	: REG.SEL,*	
11499	041050	012746	000012		MOV #12,-(SP)		
11500	041054	060616			ADD SP,(SP)	: INDEX,*	
11501	041056	004767	162012		JSR PC,WRT.REG		
11502	041062	010316			MOV R3,(SP)	: TST.PAT,*	5863
11503	041064	010246			MOV R2,-(SP)	: REG.SEL,*	
11504	041066	012746	000014		MOV #14,-(SP)		
11505	041072	060616			ADD SP,(SP)	: ERR.FLG,*	
11506	041074	004767	162362		JSR PC,RD.REG		
11507	041100	026627	000014	000001	CMP 14(SP),#1	: ERR.FLG,*	5865
11508	041106	001054			BNE 78		
11509	041110	104455			TRAP 55	:	5868
11510	041112	000172			.WORD 172		
11511	041114	010570			.WORD ARR.DAT		
11512	041116	024052			.WORD DUMPER		
11513	041120	012746	007122		MOV #WORD.56,-(SP)	:	5869
11514	041124	012746	007066		MOV #WORD.52,-(SP)		
11515	041130	012746	010004		MOV #FNC.6,-(SP)		
11516	041134	012746	007774		MOV #FNC.5,-(SP)		
11517	041140	012746	006426		MOV #WORD.12,-(SP)		
11518	041144	012746	007454		MOV #PHR.4,-(SP)		
11519	041150	012746	006100		MOV #SIX.FMT,-(SP)		
11520	041154	012746	000007		MOV #7,-(SP)		
11521	041160	010600			MOV SP,RO	: SP,*	
11522	041162	104414			TRAP 14	:	
11523	041164	016716	151740		MOV RD.DATA,(SP)	:	5870
11524	041170	016746	151732		MOV WT.DATA,-(SP)		
11525	041174	016600	000040		MOV 40(SP),RO	: INDEX,*	
11526	041200	006300			ASL RO		
11527	041202	006300			ASL RO		
11528	041204	006300			ASL RO		
11529	041206	016046	013154		MOV ML.REG(RO),-(SP)		
11530	041212	012746	005216		MOV #FMT.16,-(SP)		
11531	041216	012746	000004		MOV #4,-(SP)		
11532	041222	010600			MOV SP,RO	: SP,*	
11533	041224	104414			TRAP 14		
11534	041226	012766	000001	000042	MOV #1,42(SP)	: *,DODU.FLG	5871
11535	041234	062706	000030		ADD #30,SP	:	5867
11536	041240	062706	000012		ADD #12,SP	:	5859
11537	041244	104467		78:	TRAP 67	:	5872
11538	041246	006000			ROR RO		
11539	041250	103660			BLO 68		

Address	Hex	Hex	Hex	Label	Op	Comments	Time	Page
11541				:ML4			22-Dec-1980 09:24:31	TOPS
11542				:			22-Dec-1980 09:21:22	PA:4
11543				:				
11544	041252	005202			INC R2	: REG.SEL		5858
11545	041254	020227	000015		CMP R2,#15	: REG.SEL,*		
11546	041260	003654			BLE 68			
11547	041262	005103			COM R3	: TST.PAT		5877
11548	041264	005201			INC R1	: TWICE		5855
11549	041266	020127	000001		CMP R1,#1	: TWICE,*		
11550	041272	003645			BLE 58			
11551	041274	006303			ASL R3	: TST.PAT		5880
11552	041276	005204			INC R4	: SHIFT		5852
11553	041300	020427	000017		CMP R4,#17	: SHIFT,*		
11554	041304	003637			BLE 48			
11555	041306	012767	000001	151622	MOV #1,REG.INIT.FLG			5888
11556	041314	012703	177777		MOV #-1,R3	: *,TST.PAT		5889
11557	041320	005004			CLR R4	: TWICE		5891
11558	041322	012702	000013		MOV #13,R2	: *,REG.SEL		5894
11559	041326	104402		88: 98:	TRAP 2			5895
11560	041330	152777	000040	151656	BISB #40,@ML.REG+40			5896
11561	041336	016701	152240		MOV ML,DUT,R1			
11562	041342	042701	177770		BIC #177770,R1			
11563	041346	142777	000007	151640	BICB #7,@ML.REG+40			
11564	041354	150177	151634		BISB R1,@ML.REG+40			
11565	041360	010346			MOV R3,-(SP)	: TST.PAT,*		5898
11566	041362	010246			MOV R2,-(SP)	: REG.SEL,*		
11567	041364	012746	000012		MOV #12,-(SP)			
11568	041370	060616			ADD SP,(SP)	: INDEX,*		
11569	041372	004767	161476		JSR PC,WRT.REG			
11570	041376	016600	000012		MOV 12(SP),R0	: INDEX,*		5899
11571	041402	006300			ASL R0			
11572	041404	006300			ASL R0			
11573	041406	006300			ASL R0			
11574	041410	010001			MOV R0,R1			
11575	041412	016105	013156		MOV ML,REG+2(R1),R5	: *,CLR.DATA		
11576	041416	056105	013162		BIS ML,REG+6(R1),R5	: *,CLR.DATA		
11577	041422	012767	000001	151506	MOV #1,REG.INIT.FLG			5900
11578	041430	010516			MOV R5,(SP)	: CLR.DATA,*		5901
11579	041432	010246			MOV R2,-(SP)	: REG.SEL,*		
11580	041434	012746	000014		MOV #14,-(SP)			
11581	041440	060616			ADD SP,(SP)	: ERR.FLG,*		
11582	041442	004767	162014		JSR PC,RD.REG			
11583	041446	026627	000014	000001	CMP 14(SP),#1	: ERR.FLG,*		5903
11584	041454	001044			BNE 108			
11585	041456	104455			TRAP 55			5906
11586	041460	000166			.WORD T66			
11587	041462	010570			.WORD ARR.DAT			
11588	041464	024052			.WORD DUMPER			
11589	041466	012746	007122		MOV #WRD.56,-(SP)			5907
11590	041472	012746	010242		MOV #FNC.23,-(SP)			
11591	041476	012746	007066		MOV #WRD.52,-(SP)			
11592	041502	012746	006426		MOV #WRD.12,-(SP)			
11593	041506	012746	007454		MOV #PHR.4,-(SP)			
11594	041512	012746	006062		MOV #FIV.FMT,-(SP)			
11595	041516	012746	000006		MOV #6,-(SP)			

```

11597          :ML4
11598          :
11599
11600 041522 010600          MOV      SP,R0          ; SP,*
11601 041524 104414          TRAP     14
11602 041526 016716 151376  MOV      RD,DATA,(SP)  ;
11603 041532 010546          MOV      R5,-(SP)      ; CLR.DATA,*
11604 041534 016146 013154  MOV      ML,REG(R1),-(SP)
11605 041540 012746 005216  MOV      #FMT.16,-(SP)
11606 041544 012746 000004  MOV      #4,-(SP)
11607 041550 010600          MOV      SP,R0          ; SP,*
11608 041552 104414          TRAP     14
11609 041554 012766 000001 000040  MOV      #1,40(SP)     ; *,DODU.FLG
11610 041562 062706 000026  ADD      #26,SP        ;
11611 041566 062706 000012 10$:    ADD      #12,SP        ;
11612 041572 104467          TRAP     67            ;
11613 041574 006000          ROR      R0
11614 041576 103653          BLO     98
11615 041600 005202          INC      R2
11616 041602 020227 000015  CMP      R2,#15        ; REG.SEL
11617 041606 003647          BLE     98             ; REG.SEL,*
11618 041610 005103          COM     R3
11619 041612 005204          INC      R4
11620 041614 020427 000001  CMP      R4,#1        ; TST.PAT
11621 041620 003640          BLE     88             ; TWICE
11622 041622 005067 151310  CLR      REG.INIT.FLG ;
11623 041626 021627 000001  CMP      (SP),#1      ; DODU.FLG,*
11624 041632 001004          BNE     11$
11625 041634 016700 151740  MOV      ML,LUN,R0    ;
11626 041640 104451          TRAP     51
11627 041642 104444          TRAP     44
11628 041644 062706 000006 11$:    ADD      #6,SP        ;
11629 041650 000207          RTS      PC
11630
11631          ; Routine Size: 312 words
11632          ; Maximum stack depth per invocation: 26 words
11637
11638
11642
11646 041652          T17::
11647 041652 004767 176614 1$:    JSR      PC,$T17      ;
11648 041656 104466          TRAP     66
11649 041660 006000          ROR      R0
11650 041662 103773          BLO     18

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

5908

5909

5905

5895

5910

5894

5915

5891

5918

5920

5923

5789

5925

11652
11653
11654
11655 041664 000207
11656
11657
11658
11663
11664
11665 :

:ML4
:

RTS PC

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

5928 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (68)

```

11667 :ML4
11668 :
11669 :
11670 : 5929 :
11671 : 5930 :
11672 : 5931 BGNSTST;
11673 : 5932 :
11674 : 5933 :
11675 : 5934 :++
11676 : 5935 : TEST NUMBER: TST 18
11677 : 5936 : TEST NAME: COMPOSIT ERROR TEST
11678 : 5937 :
11679 : 5938 : TEST DESCRIPTION:
11680 : 5939 : TEST TO SEE IF SETTING EACH
11681 : 5940 : BIT IN THE ERROR REGISTER
11682 : 5941 : CAUSES A COMPOSIT ERROR BY:
11683 : 5942 :
11684 : 5943 : WRITING A SHIFTING ONE THROUGH
11685 : 5944 : THE ERROR REGISTER (SKIPPING THE
11686 : 5945 : READ ONLY BITS) AND TESTING THE
11687 : 5946 : COMPOSIT ERROR BIT IN MLDS
11688 : 5947 : FOR BEING SET AFTER EACH
11689 : 5948 : WRITE.
11690 : 5949 :
11691 : 5950 :
11692 : 5951 :
11693 : 5952 : local
11694 : 5953 : DODU_FLG,
11695 : 5954 : DAT_PAT,
11696 : 5955 : SKIP_MASK;
11697 : 5956 :
11698 : 5957 : CLR MBUS;
11699 : 5958 : DODU_FLG = ZERO;
11700 : 5959 : SKIP_MASK = %o'163157';
11701 : 5960 : DAT_PAT = ONE;
11702 : 5961 : DODU_FLG = ZERO;
11703 : 5962 :
11704 : 5963 : incr COUNT from 0 to 15 do
11705 : 5964 : begin
11706 : 5965 :
11707 : 5966 : if (.DAT_PAT and .SKIP_MASK) neq ZERO
11708 : 5967 : then
11709 : 5968 : begin
11710 : 5969 : BGNSUB;
11711 : 5970 : MLER = .DAT_PAT;
11712 : 5971 :
11713 : 5972 : if .COMP_ERR IS_NOT_SET
11714 : 5973 : then
11715 : 5974 : begin
11716 : 5975 : ERROF (50, ASYNC, DUMPER); !ERROR IF NO COMP ERROR
11717 : 5976 : PRINTB (FOR_FMT, FNC 8, PHR_1, WRD_12, FNC_8);
11718 : 5977 : PRINTB (FMT_4, .DAT_PAT);
11719 : 5978 : DODU_FLG = ONE;
11720 : 5979 : end;
11721 : 5980 :

```

!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-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (68)

```

11723 :ML4
11724 :
11725 :
11726 :      5981      ENDSUB;
11727 :      5982      end;
11728 :
11729 :      5983
11729 :      5984      DAT_PAT = .DAT_PAT^ONE;
11730 :      5985      end;
11731 :
11732 :      5986
11732 :      5987      if .DODU_FLG IS_SET
11733 :      5988      then
11734 :      5989      begin
11735 :      5990      DODU (.ML_LUN);
11736 :      5991      DOCLN;
11737 :      5992      end;
11738 :
11739 :      5993
11739 :      5994      ENDTST;

```

!SHIFT DAT_PAT TO NEXT BIT AND REPEAT

!DROP UNIT IF DODU_FLG IS_SET

```

11747 041666 004167 142204      $T18: JSR      R1,$SAVE4      :
11748 041672 152777 000040 151314 BISB     #40,@ML.REG+40      :
11749 041700 016704 151676      MOV     ML,DUT,R4      :
11750 041704 042704 177770      BIC     #177770,R4      :
11751 041710 142777 000007 151276 BICB     #7,@ML.REG+40
11752 041716 150477 151272      BISB     R4,@ML.REG+40
11753 041722 005001      CLR     R1      : DODU.FLG      5958
11754 041724 012704 163157      MOV     #-14621,R4      : *,SKIP.MASK      5959
11755 041730 012702 000001      MOV     #1,R2      : *,DAT.PAT      5960
11756 041734 005003      CLR     R3      : COUNT      5963
11757 041736 030204      1$: BIT     R2,R4      : DAT.PAT,SKIP.MASK 5966
11758 041740 001447      BEQ     4$
11759 041742 104402      2$: TRAP    2      :
11760 041744 010277 151264      MOV     R2,@ML.REG+60      : DAT.PAT,*      5968
11761 041750 032777 040000 151246 BIT     #40000,@ML.REG+50      :
11762 041756 001035      BNE     3$
11763 041760 104455      TRAP    55      :
11764 041762 000062      .WORD   62      :
11765 041764 010464      .WORD   ASYNC
11766 041766 024052      .WORD   DUMPER
11767 041770 012746 010022      MOV     #FNC.8,-(SP)      :
11768 041774 012746 006426      MOV     #WRD.12,-(SP)
11769 042000 012746 007366      MOV     #PHR.1,-(SP)
11770 042004 012746 010022      MOV     #FNC.8,-(SP)
11771 042010 012746 006046      MOV     #FOR.FMT,-(SP)
11772 042014 012746 000005      MOV     #5,-(SP)
11773 042020 010600      MOV     SP,R0      : SP,*
11774 042022 104414      TRAP    14
11775 042024 010216      MOV     R2,(SP)      : DAT.PAT,*      5977
11776 042026 012746 004400      MOV     #FMT.4,-(SP)

```

```

11778                                     ;ML4
11779                                     ;
11780                                     ;
11781 042032 012746 000002             MOV    #2,-(SP)
11782 042036 010600                     MOV    SP,R0                ; SP,*
11783 042040 104414                     TRAP   14
11784 042042 012701 000001             MOV    #1,R1                ; *,DODU.FLG
11785 042046 062706 000020             ADD    #20,SP                ;
11786 042052 104467                     3S:   TRAP   67                ;
11787 042054 006000                     ROR    R0
11788 042056 103731                     BLO   2S
11789 042060 006302                     4S:   ASL   R2                ; DAT.PAT
11790 042062 005203                     INC    R3                    ; COUNT
11791 042064 020327 000017             CMP    R3,#17                ; COUNT,*
11792 042070 003722                     BLE   1S
11793 042072 005301                     DEC    R1                    ; DODU.FLG
11794 042074 001004                     BNE   5S
11795 042076 016700 151476             MOV    ML,LUN,R0            ;
11796 042102 104451                     TRAP   51
11797 042104 104444                     TRAP   44
11798 042106 000207                     5S:   RTS    PC                ;
11799
11800                                     ; Routine Size: 73 words
11801                                     ; Maximum stack depth per invocation: 13 words
11806
11807
11811
11815 042110                                     T18::
11816 042110 004767 177552             1S:   JSR    PC,$T18          ;
11817 042114 104466                     TRAP   66
11818 042116 006000                     ROR    R0
11819 042120 103773                     BLO   1S
11820 042122 000207                     RTS    PC
11821
11822                                     ; Routine Size: 6 words
11823                                     ; Maximum stack depth per invocation: 0 words
11828
11829
11830 ;                5995 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

5978
5974
5979
5984
5963
5987
5990
5927

5992

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (69)

```

11832 :ML4
11833 :
11834 :
11835 : 5996 :
11836 : 5997 :
11837 : 5998 BGNTST:
11838 : 5999 :
11839 : 6000 :++
11840 : 6001 : TEST NUMBER: TST 19
11841 : 6002 :
11842 : 6003 : TEST NAME: ATA BIT TEST
11843 : 6004 :
11844 : 6005 : TEST DESCRIPTION:
11845 : 6006 :
11846 : 6007 : TEST THE ATA BIT FOR SETTING
11847 : 6008 : BY:
11848 : 6009 :
11849 : 6010 : 1. SETTING A BIT IN THE ERROR
11850 : 6011 : REGISTER.
11851 : 6012 :
11852 : 6013 : TEST THE ATA BIT FOR CLEARING
11853 : 6014 : AFTER BEING SET BY:
11854 : 6015 : 1. WRITING A FUNCTION TO MLCS1.
11855 : 6016 : 2. WRITING A ONE INTO THIS
11856 : 6017 : UNITS ATA BIT
11857 : 6018 :
11858 : 6019 : TEST THE ATA BIT FOR NOT CLEARING
11859 : 6020 : AFTER BEING SET BY:
11860 : 6021 :
11861 : 6022 : 1. WRITING A ONE INTO THE
11862 : 6023 : ATA BIT OF THE OTHER
11863 : 6024 : UNITS.
11864 : 6025 :
11865 : 6026 :
11866 : 6027 :
11867 : 6028 local
11868 : 6029 ATA_SAVE : bitvector [8],
11869 : 6030 DAT_PAT;
11870 : 6031
11871 : 6032 CLR MBUS;
11872 : 6033 MLER = ONE;
11873 : 6034 MLER = ZERO;
11874 : 6035 ATA_SAVE = .MLAS;
11875 : 6036
11876 : 6037 if .ATA_SAVE [.ML_DUT] IS_NOT_SET
11877 : 6038 then
11878 : 6039 begin
11879 : 6040 ERRDF (S1, ASYNC, DUMPER);
11880 : 6041 PRINTB (FOR_FMT, WRD_15, PHR_1, WRD_11, FNC_8);
11881 : 6042 EXIT_TST;
11882 : 6043 end;
11883 : 6044
11884 : 6045 if .ATTN IS_NOT_SET
11885 : 6046 then
11886 : 6047 begin

```

!STORES ALL 8 ATA BITS ON READS AND WRITES
!DATA PATTERN

!SET THE ATA BIT

!READ THE ATTN REGISTER

!SEE IF THIS DRIVES ATA BIT IS SET

!ERROR AND EXIT_TST IF NOT SET

!SEE IF THE ATTN BIT IS SET

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (69)

```

11888 :ML4
11889 :
11890 :
11891 :      6048      ERRDF (52, ASYNC, DUMPER);          !ERROR AND EXIT_TST IF NOT SET
11892 :      6049      PRINTB (FIV_FMT, REG_2, WRD_16, PHR_1, WRD_11, FNC_8);
11893 :      6050      EXIT_TST;
11894 :      6051      end;
11895 :      6052
11896 :      6053      MLCS1 = NOOP;                          !TRY TO CLEAR THE ATA BIT WITH NOOP FUNC
11897 :      6054
11898 :      6055      if .ATTN IS_SET                          !SEE IF ATA GOT CLEARED
11899 :      6056      then
11900 :      6057      begin
11901 :      6058      ERRDF (53, ASYNC, DUMPER);          !ERROR AND EXIT_TST IF SET
11902 :      6059      PRINTB (FOR_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
11903 :      6060      EXIT_TST;
11904 :      6061      end;
11905 :      6062
11906 :      6063      ATA_SAVE = .MLAS;                          !READ THE ATTENTION REGISTER
11907 :      6064
11908 :      6065      if .ATA_SAVE [.ML_DUT] IS_SET          !SEE IF THE ATA REG GOT CLEARED BY NO-OP
11909 :      6066      then
11910 :      6067      begin
11911 :      6068      ERRDF (58, ASYNC, DUMPER);
11912 :      6069      PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
11913 :      6070      end;
11914 :      6071
11915 :      6072      BGNSUB;
11916 :      6073      MLER = ONE;                                !SET THE ATA BIT
11917 :      6074      MLER = ZERO;
11918 :      6075      ATA_SAVE = ZEROES;                        !CLEAR ATA_SAVE
11919 :      6076      ATA_SAVE [.ML_DUT] = ONE;              !SET ATA_SAVE FOR THIS DRIVE
11920 :      6077      MLAS = .ATA_SAVE;                       !TRY TO CLEAR THE ATA BY WRITING TO IT.
11921 :      6078
11922 :      6079      if .ATTN IS_SET                          !SEE IF THE ATA GOT CLEARED
11923 :      6080      then
11924 :      6081      begin
11925 :      6082      ERRDF (54, ASYNC, DUMPER);          !ERROR IF NOT CLEARED
11926 :      6083      PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, WRD_17, REG_5);
11927 :      6084      end;
11928 :      6085
11929 :      6086      ENDSUB;
11930 :      6087      BGNSUB;
11931 :      6088      MLER = ONE;                                !SET THE ATA BIT
11932 :      6089      MLER = ZERO;
11933 :      6090      DAT_PAT = ONE;                            !DATA PATTERN OF ONE IN FIELD OF ZEROES
11934 :      6091
11935 :      6092      incr ATA_SEL from 0 to 7 do              !REPEAT LOOP 8 TIMES
11936 :      6093      begin
11937 :      6094
11938 :      6095      if .ATA_SEL neq .ML_DUT                      !SKIP IF ATA_SEL EQLS THIS DRIVE NO.
11939 :      6096      then
11940 :      6097      begin
11941 :      6098      MLAS = .DAT_PAT;                            !WRITE DAT PAT TO ATA REGISTER
11942 :      6099      ATA_SAVE = .MLAS;                        !READ ATA REG BACK

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (69)

11944 :ML4
11945 :
11946 :
11947 : 6100
11948 : 6101
11949 : 6102
11950 : 6103
11951 : 6104
11952 : 6105
11953 : 6106
11954 : 6107
11955 : 6108
11956 : 6109
11957 : 6110
11958 : 6111
11959 : 6112
11960 : 6113
11961 : 6114
11962 : 6115
11963 : 6116

```

if .ATA_SAVE [.ML_DUT] IS_NOT_SET      !SEE IF THIS DRIVE ATA IS CLEARED
then
begin
ERRDF (55, ASYNC, DUMPER);             !ERROR AND EXIT LOOP IF CLEARED
PRINTB (SIX_FMT, WRD_15, PHR_6, WRD_11, WRD_17, REG_5, PHR_7);
PRINTB (FMT_7, .DAT_PAT);
exitloop;
end;
end;

DAT_PAT = .DAT_PAT^ONE;                !SHIFT DAT_PAT AND REPEAT
end;

```

ENDSUB;
ENDTST;

11971	042124	004167	141730	ST19:	JSR	R1,SSAVE3	:	5994
11972	042130	152777	000040	151056	BISB	#40,@ML.REG+40	:	6030
11973	042136	016703	151440		MOV	ML_DUT,R3	:	
11974	042142	042703	177770		BIC	#177770,R3	:	
11975	042146	142777	000007	151040	BICB	#7,@ML.REG+40	:	
11976	042154	150377	151034		BISB	R3,@ML.REG+40	:	
11977	042160	012777	000001	151046	MOV	#1,@ML.REG+60	:	6033
11978	042166	005077	151042		CLR	@ML.REG+60	:	6034
11979	042172	017746	151046		MOV	@ML.REG+70,-(SP)	:	6035
11980	042176	016701	151400		MOV	ML_DUT,R1	:	6037
11981	042202	006201			ASR	R1	:	
11982	042204	006201			ASR	R1	:	
11983	042206	006201			ASR	R1	:	
11984	042210	010600			MOV	SP,R0	:	ATA.SAVE,*
11985	042212	060001			ADD	R0,R1	:	
11986	042214	010146			MOV	R1,-(SP)	:	
11987	042216	016746	151360		MOV	ML_DUT,-(SP)	:	
11988	042222	042716	177770		BIC	#177770,(SP)	:	
11989	042226	012746	000001		MOV	#1,-(SP)	:	
11990	042232	005046			CLR	-(SP)	:	
11991	042234	004767	140700		JSR	PC,BLSGT2	:	
11992	042240	062706	000010		ADD	#10,SP	:	
11993	042244	005700			TST	R0	:	
11994	042246	001026			BNE	1\$:	
11995	042250	104455			TRAP	55	:	6040
11996	042252	000063			.WORD	63	:	
11997	042254	010464			.WORD	ASYNC	:	

```

11999          :ML4
12000          :
12001          :
12002 042256 024052          .WORD DUMPER
12003 042260 012746 010022 MOV #FNC.8,-(SP)
12004 042264 012746 006416 MOV #WORD.11,-(SP)
12005 042270 012746 007366 MOV #PHR.1,-(SP)
12006 042274 012746 006454 MOV #WORD.15,-(SP)
12007 042280 012746 006046 MOV #FOR.FMT,-(SP)
12008 042284 012746 000005 MOV #5,-(SP)
12009 042290 010600 MOV SP,R0
12010 042292 104414 TRAP 14
12011 042294 104463 TRAP 63
12012 042296 062706 000014 ADD #14,SP
12013 042298 000467 BR 4$
12014 042300 032777 100000 150672 1$: BIT #100000,@ML.REG+50
12015 042302 001026 BNE 2$
12016 042304 104455 TRAP 55
12017 042306 000064 .WORD 64
12018 042308 010464 .WORD ASYNC
12019 042310 024052 .WORD DUMPER
12020 042312 012746 010022 MOV #FNC.8,-(SP)
12021 042314 012746 006416 MOV #WORD.11,-(SP)
12022 042316 012746 007366 MOV #PHR.1,-(SP)
12023 042318 012746 006462 MOV #WORD.16,-(SP)
12024 042320 012746 010266 MOV #REG.2,-(SP)
12025 042322 012746 006062 MOV #FIV.FMT,-(SP)
12026 042324 012746 000006 MOV #6,-(SP)
12027 042400 010600 MOV SP,R0
12028 042402 104414 TRAP 14
12029 042404 104463 TRAP 63
12030 042406 000433 BR 3$
12031 042410 012777 000001 150536 2$: MOV #1,@ML.REG
12032 042412 005777 150602 TST @ML.REG+50
12033 042414 100031 BPL 5$
12034 042416 104455 TRAP 55
12035 042418 000065 .WORD 65
12036 042420 010464 .WORD ASYNC
12037 042422 024052 .WORD DUMPER
12038 042424 012746 006506 MOV #WORD.19,-(SP)
12039 042426 012746 007742 MOV #FNC.2,-(SP)
12040 042428 012746 006416 MOV #WORD.11,-(SP)
12041 042430 012746 007404 MOV #PHR.2,-(SP)
12042 042432 012746 006454 MOV #WORD.15,-(SP)
12043 042434 012746 006046 MOV #FOR.FMT,-(SP)
12044 042436 012746 000006 MOV #6,-(SP)
12045 042438 010600 MOV SP,R0
12046 042440 104414 TRAP 14
12047 042442 104463 TRAP 63
12048 042444 062706 000016 3$: ADD #16,SP
12049 042446 000167 000546 4$: JMP 12$
12050 042448 017716 150532 5$: MOV @ML.REG+70,(SP)
12051 042450 016701 151064 MOV ML.DUT,R1
12052 042452 006201 ASR R1
12053 042454 006201 ASR R1

```

6041
6037
6039
6045
6048
6049
6045
6053
6055
6058
6059
6055
6057
6063
6065

Address	Hex	Hex	Hex	Label	Comment	Address
12055						
12056						
12057						
12058	042522	006201		ASR	R1	
12059	042524	010600		MOV	SP,R0	: ATA.SAVE,*
12060	042526	060001		ADD	R0,R1	
12061	042530	010146		MOV	R1,-(SP)	
12062	042532	016746	151044	MOV	ML.DUT,-(SP)	
12063	042536	042716	177770	BIC	#177770,(SP)	
12064	042542	012746	000001	MOV	#1,-(SP)	
12065	042546	005046		CLR	-(SP)	
12066	042550	004767	140364	JSR	PC,BLSGT2	
12067	042554	062706	000010	ADD	#10,SP	
12068	042560	005300		DEC	R0	
12069	042562	001026		BNE	68	
12070	042564	104455		TRAP	55	: 6068
12071	042566	000072		.WORD	72	
12072	042570	010464		.WORD	ASYN	
12073	042572	024052		.WORD	DUMPER	
12074	042574	012746	006506	MOV	#WRD.19,-(SP)	: 6069
12075	042600	012746	007742	MOV	#FNC.2,-(SP)	
12076	042604	012746	006416	MOV	#WRD.11,-(SP)	
12077	042610	012746	007404	MOV	#PHR.2,-(SP)	
12078	042614	012746	006454	MOV	#WRD.15,-(SP)	
12079	042620	012746	006062	MOV	#FIV.FMT,-(SP)	
12080	042624	012746	000006	MOV	#6,-(SP)	
12081	042630	010600		MOV	SP,R0	: SP,*
12082	042632	104414		TRAP	14	
12083	042634	062706	000016	ADD	#16,SP	: 6067
12084	042640	104402		TRAP	2	: 6070
12085	042642	012777	000001 150364	MOV	#1,@ML.REG+60	: 6073
12086	042650	005077	150360	CLR	@ML.REG+60	: 6074
12087	042654	005016		CLR	(SP)	: 6075
12088	042656	016701	150720	MOV	ML.DUT,R1	: 6076
12089	042662	006201		ASR	R1	
12090	042664	006201		ASR	R1	
12091	042666	006201		ASR	R1	
12092	042670	010600		MOV	SP,R0	: ATA.SAVE,*
12093	042672	060001		ADD	R0,R1	
12094	042674	010146		MOV	R1,-(SP)	
12095	042676	016746	150700	MOV	ML.DUT,-(SP)	
12096	042702	042716	177770	BIC	#177770,(SP)	
12097	042706	012746	000001	MOV	#1,-(SP)	
12098	042712	011646		MOV	(SP),-(SP)	
12099	042714	004767	140456	JSR	PC,BLSPU2	
12100	042720	016677	000010 150316	MOV	10(SP),@ML.REG+70	: ATA.SAVE,*
12101	042726	005777	150272	TST	@ML.REG+50	: 6077
12102	042732	100026		BPL	78	: 6079
12103	042734	104455		TRAP	55	: 6082
12104	042736	000066		.WORD	66	
12105	042740	010464		.WORD	ASYN	
12106	042742	024052		.WORD	DUMPER	
12107	042744	012746	010310	MOV	#REG.5,-(SP)	: 6083
12108	042750	012746	006470	MOV	#WRD.17,-(SP)	
12109	042754	012746	006416	MOV	#WRD.11,-(SP)	

Address	OpCode	OpCode	OpCode	Instruction	Comments	Line No.
12111				:ML4		
12112				:		
12113				:		
12114	042760	012746	007404	MOV #PHR.2, -(SP)		
12115	042764	012746	006454	MOV #WRD.15, -(SP)		
12116	042770	012746	006062	MOV #FIV.FMT, -(SP)		
12117	042774	012746	000006	MOV #6, -(SP)		
12118	043000	010600		MOV SP, R0	: SP, *	
12119	043002	104414		TRAP 14		
12120	043004	062706	000016	ADD #16, SP		6081
12121	043010	062706	000010	ADD #10, SP		6070
12122	043014	104467		TRAP 67		6084
12123	043016	006000		ROR R0		
12124	043020	103707		BLO 6\$		
12125	043022	104402		TRAP 2		6086
12126	043024	012777	000001	MOV #1, @ML.REG+60		6088
12127	043032	005077	150176	CLR @ML.REG+60		6089
12128	043036	012703	000001	MOV #1, R3	: *, DAT.PAT	6090
12129	043042	005002		CLR R2	: ATA.SEL	6092
12130	043044	020267	150532	CMP R2, ML.DUT	: ATA.SEL, *	6095
12131	043050	001471		BEQ 10\$		
12132	043052	010377	150166	MOV R3, @ML.REG+70	: DAT.PAT, *	6098
12133	043056	017716	150162	MOV @ML.REG+70, (SP)	: *, ATA.SAVE	6099
12134	043062	016701	150514	MOV ML.DUT, R1		6101
12135	043066	006201		ASR R1		
12136	043070	006201		ASR R1		
12137	043072	006201		ASR R1		
12138	043074	010600		MOV SP, R0	: ATA.SAVE, *	
12139	043076	060001		ADD R0, R1		
12140	043100	010146		MOV R1, -(SP)		
12141	043102	016746	150474	MOV ML.DUT, -(SP)		
12142	043106	042716	177770	BIC #177770, (SP)		
12143	043112	012746	000001	MOV #1, -(SP)		
12144	043116	005046		CLR -(SP)		
12145	043120	004767	140014	JSR PC, BLSGT2		
12146	043124	062706	000010	ADD #10, SP		
12147	043130	005700		TST R0		
12148	043132	001040		BNE 10\$		
12149	043134	104455		TRAP 55		6104
12150	043136	000067		.WORD 67		
12151	043140	010464		.WORD ASYNC		
12152	043142	024052		.WORD DUMPER		
12153	043144	012746	007516	MOV #PHR.7, -(SP)		6105
12154	043150	012746	010310	MOV #REG.5, -(SP)		
12155	043154	012746	006470	MOV #WRD.17, -(SP)		
12156	043160	012746	006416	MOV #WRD.11, -(SP)		
12157	043164	012746	007504	MOV #PHR.6, -(SP)		
12158	043170	012746	006454	MOV #WRD.15, -(SP)		
12159	043174	012746	006100	MOV #SIX.FMT, -(SP)		
12160	043200	012746	000007	MOV #7, -(SP)		
12161	043204	010600		MOV SP, R0	: SP, *	
12162	043206	104414		TRAP 14		
12163	043210	010316		MOV R3, (SP)	: DAT.PAT, *	6106
12164	043212	012746	004562	MOV #FMT.7, -(SP)		
12165	043216	012746	000002	MOV #2, -(SP)		

```
12167 ;ML4
12168 ;
12169 ;
12170 043222 010600 MOV SP,R0 ; SP,*
12171 043224 104414 TRAP 14 ;
12172 043226 062706 000024 ADD #24,SP ;
12173 043232 000405 BR 11$ ;
12174 043234 006303 10$: ASL R3 ; DAT.PAT 6107
12175 043236 005202 INC R2 ; ATA.SEL 6112
12176 043240 020227 000007 CMP R2,#7 ; ATA.SEL,* 6092
12177 043244 003677 BLE 9$ ;
12178 043246 104467 11$: TRAP 67 ; 6113
12179 043250 006000 ROR R0 ;
12180 043252 103663 BLO 8$ ;
12181 043254 005726 12$: TST (SP)+ ;
12182 043256 000207 RTS PC ; 5994
12183 ;
12184 ; Routine Size: 302 words
12185 ; Maximum stack depth per invocation: 16 words
12190 ;
12191 ;
12195 ;
12199 043260 T19::
12200 043260 004767 176640 1$: JSR PC,$T19 ;
12201 043264 104466 TRAP 66 ;
12202 043266 006000 ROR R0 ;
12203 043270 103773 BLO 1$ ;
12204 043272 000207 RTS PC ;
12205 ;
12206 ; Routine Size: 6 words
12207 ; Maximum stack depth per invocation: 0 words
12212 ;
12213 ;
12214 ; 6117 !<BLF/PAGE>
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (70)

```

12216 :ML4
12217 :
12218 :
12219 :      6118  |
12220 :      6119  |
12221 :      6120  | BGNTST:
12222 :      6121  |
12223 :      6122  | ++
12224 :      6123  | TEST NUMBER: TST 20
12225 :      6124  |
12226 :      6125  | TEST NAME: SEARCH FUNCTION TEST
12227 :      6126  |
12228 :      6127  | TEST DESCRIPTION:
12229 :      6128  |
12230 :      6129  | TEST THE SEARCH FUNCTION BY:
12231 :      6130  |
12232 :      6131  | 1. DOING A SEARCH FUNCTION AT
12233 :      6132  | ARRAY ZERO AND TEST GO,
12234 :      6133  | ERROR BITS AND ATTN FOR
12235 :      6134  | SETTING/NOT SETTING.
12236 :      6135  |
12237 :      6136  | 2. DOING SEARCH FUNCTIONS AT ALL
12238 :      6137  | PRESENT ARRAYS' AND TEST ATTN
12239 :      6138  | SET
12240 :      6139  |
12241 :      6140  | 3. DOING SEARCH FUNCTIONS AT ALL
12242 :      6141  | NOT PRESENT ARRAYS' AND TEST
12243 :      6142  | ATTN CLEARED.
12244 :      6143  |
12245 :      6144  |
12246 :      6145  |
12247 :      6146  | CLR MBUS:
12248 :      6147  | MLDA = ZEROES;
12249 :      6148  | MLCS1 = SEARCH;
12250 :      6149  |
12251 :      6150  | if .GO IS_SET
12252 :      6151  | then
12253 :      6152  | begin
12254 :      6153  | ERRDF (56, ASYNC, DUMPER);
12255 :      6154  | PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_10, WRD_19);
12256 :      6155  | end;
12257 :      6156  |
12258 :      6157  | if .ILF IS_SET
12259 :      6158  | then
12260 :      6159  | begin
12261 :      6160  | ERRDF (57, ASYNC, DUMPER);
12262 :      6161  | PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, FNC_10, WRD_19);
12263 :      6162  | end;
12264 :      6163  |
12265 :      6164  | if .ATTN IS_NOT_SET
12266 :      6165  | then
12267 :      6166  | begin
12268 :      6167  | ERRDF (59, ASYNC, DUMPER);
12269 :      6168  | PRINTB (FIV_FMT, WRD_16, PHR_1, WRD_12, FNC_10, WRD_19);
12270 :      6169  | end

```

!DO A SEARCH FUNCTION

!SEE IF GO IS SET

!ERROR IF NOT SET

!SEE IF ILF IS SET

!ERROR IF SET

!SEE IF ATTN IS SET

!ERROR IF NOT SET

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (70)

```

12272 :ML4
12273 :
12274 :
12275 :      6170 else
12276 :      6171 begin
12277 :      6172
12278 :      6173   incr ARR_SEL from 0 to .LST_ARR by .ARR_INC do      !DO SEARCH AT ALL PRESENT ARRAYS
12279 :      6174   begin
12280 :      6175   BGN SUB;
12281 :      6176   CLR MBUS;
12282 :      6177   MLDA = .ARR_SEL;      !LOAD DSA REG WITH ARR_SEL
12283 :      6178   MLCS1 = SEARCH;      !DO A SEARCH FUNCTION
12284 :      6179
12285 :      6180   if .OPI IS_SET      !READ ATTN
12286 :      6181   then
12287 :      6182   begin
12288 :      6183   ERRDF (60, ASYNC, DUMPER);      !ERROR IF NOT SET
12289 :      6184   PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_10, WRD_19);
12290 :      6185   PRINTB (FMT_9, .ARR_SEL);
12291 :      6186   end;
12292 :      6187
12293 :      6188   ENDSUB;
12294 :      6189   end;
12295 :      6190
12296 :      6191   if .OP_NUM_ARR lss %o'000017'      !SEE IF LSS 17 ARRAYS ARE PRESENT
12297 :      6192   then
12298 :      6193
12299 :      6194   incr ARR_SEL from .LST_ARR + .ARR_INC to .ARR_16 by .ARR_INC do
12300 :      6195   !DO A SEARCH AT ALL NOT PRESENT
12301 :      6196   !ARRAYS IF LSS 17
12302 :      6197   begin
12303 :      6198   BGN SUB;
12304 :      6199   CLR MBUS;
12305 :      6200   MLDA = .ARR_SEL;      !LOAD DSA REG WITH ARR_SEL
12306 :      6201   MLCS1 = SEARCH;      !DO A SEARCH FUNCTION
12307 :      6202
12308 :      6203   if .OPI IS_NOT_SET      !SEE IF OPI IS SET
12309 :      6204   then
12310 :      6205   begin
12311 :      6206   ERRDF (61, ASYNC, DUMPER);      !ERROR IF NOT SET
12312 :      6207   PRINTB (FIV_FMT, WRD_4, PHR_1, WRD_12, FNC_10, WRD_19);
12313 :      6208   PRINTB (FMT_9, .ARR_SEL);
12314 :      6209   end;
12315 :      6210   ENDSUB;
12316 :      6211   end;
12317 :      6212
12318 :      6213   end;
12319 :      6214
12320 :      6215   ENDTST;
12324 :

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

SEQ 0275

12332	043274	004167	140560		ST20:	JSR	R1,SSAVE3	:	6116
12333	043300	152777	000040	147706		BISB	#40,AML.REG+40	:	6120
12334	043306	016703	150270			MOV	ML,DUT,R3	:	
12335	043312	042703	177770			BIC	#177770,R3	:	
12336	043316	142777	000007	147670		BICB	#7,AML.REG+40	:	
12337	043324	150377	147664			BISB	R3,AML.REG+40	:	
12338	043330	005077	147650			CLR	AML.REG+30	:	6147
12339	043334	012777	000031	147612		MOV	#31,AML.REG	:	6148
12340	043342	132777	000001	147604		BITB	#1,AML.REG	:	6150
12341	043350	001426				BEQ	18	:	
12342	043352	104455				TRAP	55	:	6153
12343	043354	000070				.WORD	70	:	
12344	043356	010464				.WORD	ASYN	:	
12345	043360	024052				.WORD	DUMPER	:	
12346	043362	012746	006506			MOV	#WRD.19,-(SP)	:	6154
12347	043366	012746	010050			MOV	#FNC.10,-(SP)	:	
12348	043372	012746	006416			MOV	#WRD.11,-(SP)	:	
12349	043376	012746	007404			MOV	#PHR.2,-(SP)	:	
12350	043402	012746	006276			MOV	#WRD.1,-(SP)	:	
12351	043406	012746	006062			MOV	#FIV.FMT,-(SP)	:	
12352	043412	012746	000006			MOV	#6,-(SP)	:	
12353	043416	010600				MOV	SP,R0	: SP,*	
12354	043420	104414				TRAP	14	:	
12355	043422	062706	000016			ADD	#16,SP	:	6152
12356	043426	132777	000001	147600	18:	BITB	#1,AML.REG+60	:	6157
12357	043434	001430				BEQ	28	:	
12358	043436	104455				TRAP	55	:	6160
12359	043440	000071				.WORD	71	:	
12360	043442	010464				.WORD	ASYN	:	
12361	043444	024052				.WORD	DUMPER	:	
12362	043446	012746	006506			MOV	#WRD.19,-(SP)	:	6161
12363	043452	012746	010050			MOV	#FNC.10,-(SP)	:	
12364	043456	012746	010060			MOV	#FNC.11,-(SP)	:	
12365	043462	012746	006416			MOV	#WRD.11,-(SP)	:	
12366	043466	012746	007472			MOV	#PHR.5,-(SP)	:	
12367	043472	012746	006314			MOV	#WRD.3,-(SP)	:	
12368	043476	012746	006062			MOV	#FIV.FMT,-(SP)	:	
12369	043502	012746	000007			MOV	#7,-(SP)	:	
12370	043506	010600				MOV	SP,R0	: SP,*	
12371	043510	104414				TRAP	14	:	
12372	043512	062706	000020			ADD	#20,SP	:	6159
12373	043516	032777	100000	147500	28:	BIT	#100000,AML.REG+50	:	6164
12374	043524	001027				BNE	38	:	
12375	043526	104455				TRAP	55	:	6167
12376	043530	000073				.WORD	73	:	
12377	043532	010464				.WORD	ASYN	:	
12378	043534	024052				.WORD	DUMPER	:	
12379	043536	012746	006506			MOV	#WRD.19,-(SP)	:	6168
12380	043542	012746	010050			MOV	#FNC.10,-(SP)	:	
12381	043546	012746	006426			MOV	#WRD.12,-(SP)	:	

```

12383      :ML4
12384      :
12385
12386 043552 012746 007366      MOV      #PHR.1,-(SP)
12387 043556 012746 006462      MOV      #WRD.16,-(SP)
12388 043562 012746 006062      MOV      #FIV.FMT,-(SP)
12389 043566 012746 000006      MOV      #6,-(SP)
12390 043572 010600                MOV      SP,R0          : SP,*
12391 043574 104414                TRAP     14
12392 043576 062706 000016      ADD      #16,SP        :
12393 043602 000207                RTS      PC             :
12394 043604 016702 145766      3$:     MOV      LST.ARR,R2  :
12395 043610 016703 145746      MOV      ARR.INC,R3    :
12396 043614 005001                CLR      R1            : ARR.SEL
12397 043616 000467                BR       6$
12398 043620 104402                4$:     TRAP     2
12399 043622 152777 000040 147364      BISB    #40,@ML.REG+40 :
12400 043630 016700 147746      MOV      ML,DUT,R0    :
12401 043634 042700 177770      BIC     #177770,R0
12402 043640 142777 000007 147346      BICB    #7,@ML.REG+40
12403 043646 150077 147342      BISB    R0,@ML.REG+40
12404 043652 010177 147326      MOV      R1,@ML.REG+30 : ARR.SEL,*
12405 043656 012777 000031 147270      MOV      #31,@ML.REG  :
12406 043664 032777 020000 147342      BIT     #20000,@ML.REG+60 :
12407 043672 001435                BEQ     5$
12408 043674 104455                TRAP     55
12409 043676 000074                .WORD   74
12410 043700 010464                .WORD   ASYNC
12411 043702 024052                .WORD   DUMPER
12412 043704 012746 006506      MOV      #WRD.19,-(SP) :
12413 043710 012746 010050      MOV      #FNC.10,-(SP)
12414 043714 012743 006426      MOV      #WRD.12,-(SP)
12415 043720 012746 007472      MOV      #PHR.5,-(SP)
12416 043724 012746 006322      MOV      #WRD.4,-(SP)
12417 043730 012746 006062      MOV      #FIV.FMT,-(SP)
12418 043734 012746 000006      MOV      #6,-(SP)
12419 043740 010600                MOV      SP,R0          : SP,*
12420 043742 104414                TRAP     14
12421 043744 010116      MOV      R1,(SP)      : ARR.SEL,*
12422 043746 012746 004644      MOV      #FMT.9,-(SP) :
12423 043752 012746 000002      MOV      #2,-(SP)
12424 043756 010600                MOV      SP,R0          : SP,*
12425 043760 104414                TRAP     14
12426 043762 062706 000022      ADD      #22,SP        :
12427 043766 104467                5$:     TRAP     67
12428 043770 006000                ROR     R0
12429 043772 103712                BLO     4$
12430 043774 060301                ADD     R3,R1          : *,ARR.SEL
12431 043776 020102                CMP     R1,R2          : ARR.SEL,*
12432 044000 003707                BLE     4$
12433 044002 026727 145552 000017      CMP     OP.NUM.ARR,#17 :
12434 044010 002102                BGE     10$
12435 044012 016701 145560      MOV      LST.ARR,R1   :
12436 044016 066701 145540      ADD     ARR.INC,R1
12437 044022 016703 145546      MOV     ARR.16,R3

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 290⁶

SEQ 0277

12439									22-Dec-1980 09:24:31	TOPS
12440									22-Dec-1980 09:21:22	PA: <
12441										
12442	044026	016702	145530		MOV	ARR.INC,R2				
12443	044032	000467			BR	9\$				
12444	044034	104402		7\$:	TRAP	2	:			6196
12445	044036	152777	000040	147150	BISB	#40,@ML.REG+40	:			6197
12446	044044	016700	147532		MOV	ML,DUT,R0				
12447	044050	042700	177770		BIC	#177770,R0				
12448	044054	142777	000007	147132	BICB	#7,@ML.REG+40				
12449	044062	150077	147126		BISB	R0,@ML.REG+40				
12450	044066	010177	147112		MOV	R1,@ML.REG+30	:	ARR.SEL,*		6199
12451	044072	012777	000031	147054	MOV	#31,@ML.REG	:			6200
12452	044100	032777	020000	147126	BIT	#20000,@ML.REG+60	:			6202
12453	044106	001035			BNE	8\$:			
12454	044110	104455			TRAP	55	:			6205
12455	044112	000075			.WORD	75				
12456	044114	010464			.WORD	ASYN				
12457	044116	024052			.WORD	DUMPER				
12458	044120	012746	006506		MOV	#WRD.19,-(SP)	:			6206
12459	044124	012746	010050		MOV	#FNC.10,-(SP)				
12460	044130	012746	006426		MOV	#WRD.12,-(SP)				
12461	044134	012746	007366		MOV	#PHR.1,-(SP)				
12462	044140	012746	006322		MOV	#WRD.4,-(SP)				
12463	044144	012746	006062		MOV	#FIV.FMT,-(SP)				
12464	044150	012746	000006		MOV	#6,-(SP)				
12465	044154	010600			MOV	SP,R0	:	SP,*		
12466	044156	104414			TRAP	14	:			
12467	044160	010116			MOV	R1,(SP)	:	ARR.SEL,*		6207
12468	044162	012746	004644		MOV	#FMT.9,-(SP)				
12469	044166	012746	000002		MOV	#2,-(SP)				
12470	044172	010600			MOV	SP,R0	:	SP,*		
12471	044174	104414			TRAP	14				
12472	044176	062706	000022		ADD	#22,SP	:			6204
12473	044202	104467		8\$:	TRAP	67	:			6208
12474	044204	006000			ROR	R0				
12475	044206	103712			BLO	7\$				
12476	044210	060201			ADD	R2,R1	:	*,ARR.SEL		6194
12477	044212	020103		9\$:	CMP	R1,R3	:	ARR.SEL,*		
12478	044214	003707			BLE	7\$				
12479	044216	000207		10\$:	RTS	PC	:			6116
12480										
12481										
12482										
12487										
12488										
12492										

: Routine Size: 234 words
: Maximum stack depth per invocation: 13 words

12494
12495
12496
12500 044220
12501 044220 004767 177050
12502 044224 104466
12503 044226 006000
12504 044230 103773
12505 044232 000207
12506
12507
12508
12513
12514
12515 ; 6216 !<BLF/PAGE>

;ML4
;
T20::
1\$: JSR PC,\$T20
 TRAP 66
 ROR R0
 BLO 1\$
 RTS PC

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

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

6215

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (71)

```

12517 :ML4
12518 :
12519 :
12520 : 6217 !
12521 : 6218 ! BGNTST;
12522 : 6219 !
12523 : 6220 ! ++
12524 : 6221 ! TEST NUMBER: TST 21
12525 : 6222 !
12526 : 6223 ! TEST NAME: READ IN PRESET TEST
12527 : 6224 !
12528 : 6225 ! TEST DESCRIPTION:
12529 : 6226 ! TEST THE READ IN PRESET FUNCTION BY:
12530 : 6227 !
12531 : 6228 ! 1. PERFORMING A READ-IN-PRESET FUNCTION AND TESTING GO, ERROR BITS
12532 : 6229 ! AND VV FOR SET / NOT SET.
12533 : 6230 !
12534 : 6231 ! -
12535 : 6232 !
12536 : 6233 CLR MBUS;
12537 : 6234 MLCS1 = RD_IN_PRE; !DO A READ IN PRESET FUNCTION
12538 : 6235 !
12539 : 6236 if .GO IS_SET !SEE IF GO IS NOT SET
12540 : 6237 then
12541 : 6238 begin
12542 : 6239 ERRDF (62, ASYNC, DUMPER); !ERROR IF SET
12543 : 6240 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_11, WRD_19);
12544 : 6241 end;
12545 : 6242 !
12546 : 6243 if .ILF IS_SET !SEE IF ILF IS NOT SET
12547 : 6244 then
12548 : 6245 begin
12549 : 6246 ERRDF (63, ASYNC, DUMPER); !ERROR IF SET
12550 : 6247 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, WRD_19);
12551 : 6248 end;
12552 : 6249 !
12553 : 6250 if .OPI IS_SET !SEE IF OPI IS NOT SET
12554 : 6251 then
12555 : 6252 begin
12556 : 6253 ERRDF (64, ASYNC, DUMPER); !ERROR IF SET
12557 : 6254 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_11, FNC_11, WRD_19);
12558 : 6255 end;
12559 : 6256 !
12560 : 6257 if .VV IS_NOT_SET !SEE IF VV IS SET
12561 : 6258 then
12562 : 6259 begin
12563 : 6260 ERRDF (65, ASYNC, DUMPER); !ERROR IF NOT SET
12564 : 6261 PRINTB (FIV_FMT, WRD_18, PHR_1, WRD_11, FNC_11, WRD_19);
12565 : 6262 end;
12566 : 6263 !
12567 : 6264 !
12571 :

```

Address	Hex	Hex	Hex	Label	Instruction	Comment	Seq
12573				:ML4			
12574				:			
12575				:			
12579	044234	152777	000040	146752 \$T21:	BISB #40, @ML.REG+40	:	6218
12580	044242	016700	147334		MOV ML, DUT, RO	:	
12581	044246	042700	177770		BIC #177770, RO	:	
12582	044252	142777	000007	146734	BICB #7, @ML.REG+40	:	
12583	044260	150077	146730		BISB RO, @ML.REG+40	:	
12584	044264	012777	000021	146662	MOV #21, @ML.REG	:	6234
12585	044272	132777	000001	146654	BITB #1, @ML.REG	:	6236
12586	044300	001426			BEQ 1\$:	
12587	044302	104455			TRAP 55	:	6239
12588	044304	000076			.WORD 76	:	
12589	044306	010464			.WORD ASYNC	:	
12590	044310	024052			.WORD DUMPER	:	
12591	044312	012746	006506		MOV @WORD.19, -(SP)	:	6240
12592	044316	012746	010060		MOV #FNC.11, -(SP)	:	
12593	044322	012746	006416		MOV @WORD.11, -(SP)	:	
12594	044326	012746	007404		MOV #PHR.2, -(SP)	:	
12595	044332	012746	006276		MOV @WORD.1, -(SP)	:	
12596	044336	012746	006062		MOV #FIV.FMT, -(SP)	:	
12597	044342	012746	000006		MOV #6, -(SP)	:	
12598	044346	010600			MOV SP, RO	: SP,*	
12599	044350	104414			TRAP 14	:	
12600	044352	062706	000016		ADD #16, SP	:	6238
12601	044356	132777	000001	146650 1\$:	BITB #1, @ML.REG+60	:	6243
12602	044364	001426			BEQ 2\$:	
12603	044366	104455			TRAP 55	:	6246
12604	044370	000077			.WORD 77	:	
12605	044372	010464			.WORD ASYNC	:	
12606	044374	024052			.WORD DUMPER	:	
12607	044376	012746	006506		MOV @WORD.19, -(SP)	:	6247
12608	044402	012746	010060		MOV #FNC.11, -(SP)	:	
12609	044406	012746	006416		MOV @WORD.11, -(SP)	:	
12610	044412	012746	007472		MOV #PHR.5, -(SP)	:	
12611	044416	012746	006314		MOV @WORD.3, -(SP)	:	
12612	044422	012746	006062		MOV #FIV.FMT, -(SP)	:	
12613	044426	012746	000006		MOV #6, -(SP)	:	
12614	044432	010600			MOV SP, RO	: SP,*	
12615	044434	104414			TRAP 14	:	
12616	044436	062706	000016		ADD #16, SP	:	6245
12617	044442	032777	020000	146564 2\$:	BIT #20000, @ML.REG+60	:	6250
12618	044450	001426			BEQ 3\$:	
12619	044452	104455			TRAP 55	:	6253
12620	044454	000100			.WORD 100	:	
12621	044456	010464			.WORD ASYNC	:	
12622	044460	024052			.WORD DUMPER	:	
12623	044462	012746	006506		MOV @WORD.19, -(SP)	:	6254
12624	044466	012746	010060		MOV #FNC.11, -(SP)	:	
12625	044472	012746	006416		MOV @WORD.11, -(SP)	:	
12626	044476	012746	007472		MOV #PHR.5, -(SP)	:	

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

12628      ;ML4
12629      ;
12630
12631 044502 012746 006322      MOV      #WRD.4,-(SP)
12632 044506 012746 006062      MOV      #FIV.FMT,-(SP)
12633 044512 012746 000006      MOV      #6,-(SP)
12634 044516 010600              MOV      SP,R0          ; SP,*
12635 044520 104414              TRAP     14
12636 044522 062706 000016      ADD      #16,SP          ;
12637 044526 132777 000100 146470 3$: BITB     #100,@ML.REG+50 ;
12638 044534 001026              BNE      4$             ;
12639 044536 104455              TRAP     55             ;
12640 044540 000101              .WORD   101             ;
12641 044542 010464              .WORD   ASYNC           ;
12642 044544 024052              .WORD   DUMPER          ;
12643 044546 012746 006506      MOV      #WRD.19,-(SP) ;
12644 044552 012746 010060      MOV      #FNC.11,-(SP) ;
12645 044556 012746 006416      MOV      #WRD.11,-(SP) ;
12646 044562 012746 007366      MOV      #PHR.1,-(SP) ;
12647 044566 012746 006502      MOV      #WRD.18,-(SP) ;
12648 044572 012746 006062      MOV      #FIV.FMT,-(SP) ;
12649 044576 012746 000006      MOV      #6,-(SP)      ;
12650 044602 010600              MOV      SP,R0          ; SP,*
12651 044604 104414              TRAP     14
12652 044606 062706 000016      ADD      #16,SP          ;
12653 044612 000207 4$:      RTS      PC             ;
12654
12655      ; Routine Size: 120 words
12656      ; Maximum stack depth per invocation: 7 words
12661
12662
12666
12670 044614      T21::
12671 044614 004767 177414 1$:      JSR      PC,$T21        ;
12672 044620 104466              TRAP     66             ;
12673 044622 006000              ROR      R0
12674 044624 103773              BLO      1$
12675 044626 000207              RTS      PC
12676
12677      ; Routine Size: 6 words
12678      ; Maximum stack depth per invocation: 0 words

```

6252
6257

6260

6261

6259
6215

6262

+

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 295^{J 6}

SEQ 0282

12687
12688
12689 : 6265 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (72)

```

12691 :ML4
12692 :
12693 :
12694 : 6266 !
12695 : 6267 !
12696 : 6268 ! BGNTST:
12697 : 6269 !
12698 : 6270 ! ++
12699 : 6271 ! TEST NUMBER: TST 22
12700 : 6272 !
12701 : 6273 ! TEST NAME: ILLEGAL FUNCTION TEST
12702 : 6274 !
12703 : 6275 ! TEST DESCRIPTION:
12704 : 6276 !
12705 : 6277 ! TEST THE DETECTION OF ILLEGAL
12706 : 6278 ! FUNCTIONS WRITTEN TO MLCS1
12707 : 6279 ! BY:
12708 : 6280 !
12709 : 6281 ! WRITING ALL POSSIBLE ILLEGAL
12710 : 6282 ! FUNCTIONS TO MLCS1. THEN
12711 : 6283 ! TEST GO AND ERROR BITS CLEARED.
12712 : 6284 !
12713 : 6285 !
12714 : 6286 !
12715 : 6287 ! local
12716 : 6288 ! BAD_BITS, !STORES A COUNT TO GENERATE BAD FUNCTIONS
12717 : 6289 ! BAD_FUNC; !STORES GENERATED BAD FUNCTION
12718 : 6290 !
12719 : 6291 ! BAD_BITS = ZEROES; !CLEAR BAD BITS
12720 : 6292 !
12721 : 6293 ! incr CNT_1 from 0 to 2 do !REPEAT LOOP 3 TIMES
12722 : 6294 ! begin
12723 : 6295 ! BAD_BITS = .BAD_BITS + %0'2'; !ADD 2 TO BAD_BITS
12724 : 6296 !
12725 : 6297 ! incr CNT_2 from %0'1' to %0'71' by %0'10' do !REPEAT LOOP GENERATING 'GOOD' FUNCTIONS
12726 : 6298 ! begin
12727 : 6299 ! BGNSUB;
12728 : 6300 ! CLR_MBUS;
12729 : 6301 ! BAD_FUNC = .CNT_2 + .BAD_BITS; !ADD BAD BITS TO CNT_2 GENERATING BAD FUNCTIONS
12730 : 6302 ! FIRST_BLK_XFER ?); !SET UP A FIRST BLOCK XFERR
12731 : 6303 ! MLCS1 = .BAD_FUNC; !LOAD MLCS1 WITH TWO BAD FUNCTIONS
12732 : 6304 !
12733 : 6305 ! if .ILF IS_SET !SEE IF ILF IS SET
12734 : 6306 ! then
12735 : 6307 ! begin
12736 : 6308 !
12737 : 6309 ! if .GO IS_SET !IF ILF IS SET THEN SEE IF GO IS SET
12738 : 6310 ! then
12739 : 6311 ! begin
12740 : 6312 ! ERRDF (66, ASYNC, DUMPER); !ERROR IF GO SET WITH BAD FUNCTION
12741 : 6313 ! PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_12, WRD_19);
12742 : 6314 ! PRINTB (FMT_12, .BAD_FUNC);
12743 : 6315 ! end;
12744 : 6316 !
12745 : 6317 ! end

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (72)

12747 :ML4
12748 :
12749 :
12750 : 6318
12751 : 6319
12752 : 6320
12753 : 6321
12754 : 6322
12755 : 6323
12756 : 6324
12757 : 6325
12758 : 6326
12759 : 6327
12760 : 6328
12761 : 6329
12762 : 6330
12763 : 6331
12764 : 6332
12765 : 6333
12766 : 6334
12767 : 6335
12768 : 6336
12769 : 6337
12770 : 6338
12774 :

```

else
  begin
    ERRDF (67, ASYNC, DUMPER);          !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, DUMPER);          !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;

```

12778	044630	004167	137262	\$T22:	JSR	R1,\$SAVES	:	6264
12779	044634	005004			CLR	R4	:	6291
12780	044636	005001			CLR	R1	:	6293
12781	044640	062704	000002	1\$:	ADD	#2,R4	:	6295
12782	044644	012702	000001		MOV	#1,R2	:	6297
12783	044650	010405		2\$:	MOV	R4,R5	:	6301
12784	044652	060205			ADD	R2,R5	:	
12785	044654	104402		3\$:	TRAP	2	:	6298
12786	044656	152777	000040	146330	BISB	#40,@ML.REG+40	:	6299
12787	044664	016700	146712		MOV	ML.DUT,R0	:	
12788	044670	042700	177770		BIC	#177770,R0	:	
12789	044674	142777	000007	146312	BICB	#7,@ML.REG+40	:	
12790	044702	150077	146306		BISB	R0,@ML.REG+40	:	
12791	044706	010503			MOV	R5,R3	:	6301
12792	044710	004767	151124		JSR	PC,FIRST.BLK.XFER	:	6302
12793	044714	010377	146234		MOV	R3,@ML.REG	:	6303
12794	044720	132777	000001	146306	BITB	#1,@ML.REG+60	:	6305
12795	044726	001440			BEQ	4\$:	
12796	044730	132777	000001	146216	BITB	#1,@ML.REG	:	6309
12797	044736	001471			BEQ	6\$:	
12798	044740	104455			TRAP	55	:	
12799	044742	000102			.WORD	102	:	6312
12800	044744	010464			.WORD	ASYNC	:	

Address	Hex	Hex	Hex	Label	Code	Comment	Line
12802							
12803							
12804							
12805	044746	024052		.WORD	DUMPER		
12806	044750	012746	006506	MOV	#WORD.19,-(SP)	:	6313
12807	044754	012746	010100	MOV	#FNC.12,-(SP)		
12808	044760	012746	006416	MOV	#WORD.11,-(SP)		
12809	044764	012746	007404	MOV	#PHR.2,-(SP)		
12810	044770	012746	006276	MOV	#WORD.1,-(SP)		
12811	044774	012746	006062	MOV	#FIV.FMT,-(SP)		
12812	045000	012746	000006	MOV	#6,-(SP)		
12813	045004	010600		MOV	SP,R0	: SP,*	
12814	045006	104414		TRAP	14		
12815	045010	010316		MOV	R3,(SP)	: BAD.FUNC,*	6314
12816	045012	012746	005004	MOV	#FMT.12,-(SP)		
12817	045016	012746	000002	MOV	#2,-(SP)		
12818	045022	010600		MOV	SP,R0	: SP,*	
12819	045024	104414		TRAP	14		
12820	045026	000433		BR	55		6311
12821	045030	104455		TRAP	55		6320
12822	045032	000103		.WORD	103		
12823	045034	010464		.WORD	ASYN		
12824	045036	024052		.WORD	DUMPER		
12825	045040	012746	006506	MOV	#WORD.19,-(SP)	:	6321
12826	045044	012746	010100	MOV	#FNC.12,-(SP)		
12827	045050	012746	006416	MOV	#WORD.11,-(SP)		
12828	045054	012746	007366	MOV	#PHR.1,-(SP)		
12829	045060	012746	006314	MOV	#WORD.3,-(SP)		
12830	045064	012746	006062	MOV	#FIV.FMT,-(SP)		
12831	045070	012746	000006	MOV	#6,-(SP)		
12832	045074	010600		MOV	SP,R0	: SP,*	
12833	045076	104414		TRAP	14		
12834	045100	010316		MOV	R3,(SP)	: BAD.FUNC,*	6322
12835	045102	012746	005004	MOV	#FMT.12,-(SP)		
12836	045106	012746	000002	MOV	#2,-(SP)		
12837	045112	010600		MOV	SP,R0	: SP,*	
12838	045114	104414		TRAP	14		
12839	045116	062706	000022	ADD	#22,SP		6319
12840	045122	032777	020000	BIT	#20000,#ML.REG+60		6325
12841	045130	001435		BEQ	75		
12842	045132	104455		TRAP	55		6328
12843	045134	000104		.WORD	104		
12844	045136	010464		.WORD	ASYN		
12845	045140	024052		.WORD	DUMPER		
12846	045142	012746	006506	MOV	#WORD.19,-(SP)	:	6329
12847	045146	012746	010100	MOV	#FNC.12,-(SP)		
12848	045152	012746	006426	MOV	#WORD.12,-(SP)		
12849	045156	012746	007472	MOV	#PHR.5,-(SP)		
12850	045162	012746	006322	MOV	#WORD.4,-(SP)		
12851	045166	012746	006062	MOV	#FIV.FMT,-(SP)		
12852	045172	012746	000006	MOV	#6,-(SP)		
12853	045176	010600		MOV	SP,R0	: SP,*	
12854	045200	104414		TRAP	14		
12855	045202	010316		MOV	R3,(SP)	: BAD.FUNC,*	6330
12856	045204	012746	005004	MOV	#FMT.12,-(SP)		

:ML4
:

4S:

5S:
6S:

146104


```

12858      :ML4
12859      :
12860
12861 045210 012746 000002      MOV      #2,-(SP)
12862 045214 010600      MOV      SP,R0      : SP,*
12863 045216 104414      TRAP     14
12864 045220 062706 000022      ADD      #22,SP
12865 045224 104467      TRAP     67      :
12866 045226 006000      ROR      R0      :
12867 045230 103611      BLO      3$
12868 045232 062702 000010      ADD      #10,R2      : *,CNT.2
12869 045236 020227 000071      CMP      R2,#71      : CNT.2,*
12870 045242 003602      BLE      2$
12871 045244 005201      INC      R1      : CNT.1
12872 045246 020127 000002      CMP      R1,#2      : CNT.1,*
12873 045252 003002      BGT      8$
12874 045254 000167 177360      JMP
12875 045260 000207      RTS      1$
12876      :
12877      : Routine Size: 141 words
12878      : Maximum stack depth per invocation: 15 words
12883
12884
12888
12892 045262      T22::
12893 045262 004767 177342      1$: JSR      PC,$T22      :
12894 045266 104466      TRAP     66
12895 045270 006000      ROR      R0
12896 045272 103773      BLO      1$
12897 045274 000207      RTS      PC
12898
12899      : Routine Size: 6 words
12900      : Maximum stack depth per invocation: 0 words
12905
12906
12907 :      6339 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

6327
6331

6297

6293

6264

6336

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (73)

```

12909 :ML4
12910 :
12911 :
12912 : 6340 |
12913 : 6341 |
12914 : 6342 | BGNTST;
12915 : 6343 |
12916 : 6344 | ++
12917 : 6345 | TEST NUMBER: TST 23
12918 : 6346 |
12919 : 6347 | TEST NAME: REGISTER MODIFICATION REFUSED TEST
12920 : 6348 |
12921 : 6349 | TEST DESCRIPTION:
12922 : 6350 |
12923 : 6351 | TEST THE DETECTION OF A
12924 : 6352 | REGISTER MODIFICATION REFUSED
12925 : 6353 | BY:
12926 : 6354 |
12927 : 6355 | 1. WRITTING TO MLCS1, MLDA
12928 : 6356 | AND MLER WHILE THE DRIVE
12929 : 6357 | IS BUSY AND TEST RMR
12930 : 6358 | BIT SET.
12931 : 6359 |
12932 : 6360 | ALSO SEE IF THE DRIVE ASSERTED
12933 : 6361 | EXCEPTION BY TESTING THE TRE BIT SET.
12934 : 6362 |
12935 : 6363 |
12936 : 6364 |
12937 : 6365 | incr CNT from 0 to 2 do !REPEAT LOOP 3 TIMES
12938 : 6366 | begin
12939 : 6367 | BGNSUB;
12940 : 6368 | CLR MBUS;
12941 : 6369 | MLCS1 = write; !DO A WRITE FUNCTION
12942 : 6370 |
12943 : 6371 | case .CNT from 0 to 2 of !WRITE TO SELECTED REGISTERS FORCING RMR
12944 : 6372 | set
12945 : 6373 |
12946 : 6374 | [0] :
12947 : 6375 | MLCS1 = %0'000000';
12948 : 6376 |
12949 : 6377 | [1] :
12950 : 6378 | MLDA = ONES;
12951 : 6379 |
12952 : 6380 | [2] :
12953 : 6381 | MLER = ONES
12954 : 6382 | tes;
12955 : 6383 |
12956 : 6384 | DELAY (FRTY_US);
12957 : 6385 |
12958 : 6386 | if .RMR IS_NOT_SET !SEE IF RMR GOT SET
12959 : 6387 | then
12960 : 6388 | begin
12961 : 6389 | ERRDF (69, ASYNC, DUMPER); !ERROR IF NOT SET
12962 : 6390 | PRINTB (FOR_FMT, WRD_21, PHR_1, WRD_11, WRD_21);
12963 : 6391 | end;

```

22-Dec-1980 09:24:31 TOPS-20 BL1ss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (73)

```

12965 :ML4
12966 :
12967 :
12968 :      6392
12969 :      6393      if .TRE IS_NOT_SET
12970 :      6394      then
12971 :      6395      begin
12972 :      6396      ERRDF (117, SYNC, DUMPER);
12973 :      6397      PRINTB (FOR_FMT, WRD_20, PHR_1, WRD_11, WRD_21);
12974 :      6398      end;
12975 :      6399
12976 :      6400      ENDSUB;
12977 :      6401      end;
12978 :      6402
12979 :      6403      ENDTST;
12983 :

```

!SEE IF DRIVE ASSERTED EXCEPTION BY TESTING TRE

```

12987 045276 004167 136542      ST23: JSR      R1,SSAVE2      :
12988 045302 005746      TST      -(SP)          :
12989 045304 005002      CLR      R2            : CNT
12990 045306 104402      TRAP     2             :
12991 045310 152777 000040 145676      BISB     #40,@ML.REG+40 :
12992 045316 016701 146260      MOV      ML,DUT,R1     :
12993 045322 042701 177770      BIC      #177770,R1    :
12994 045326 142777 000007 145660      BICB     #7,@ML.REG+40 :
12995 045334 150177 145654      BISB     R1,@ML.REG+40 :
12996 045340 012777 000061 145606      MOV      #61,@ML.REG   :
12997 045346 010201      MOV      R2,R1        : CNT,*
12998 045350 006301      ASL      R1            :
12999 045352 066107 045356      ADD      2$(R1),PC     :
13000 045356 000006      2$:      .WORD     3$-2$   :
13001 045360 000014      .WORD     4$-2$       :
13002 045362 000024      .WORD     5$-2$       :
13003 045364 005077 145564      3$:      CLR      @ML.REG     :
13004 045370 000407      BR       6$           :
13005 045372 012777 177777 145604      4$:      MOV      #-1,@ML.REG+30 :
13006 045400 000403      BR       6$           :
13007 045402 012777 177777 145624      5$:      MOV      #-1,@ML.REG+60 :
13008 045410 012700 000050      6$:      MOV      #50,R0       : *,SSTMP2
13009 045414 001410      7$:      BEQ     10$          :
13010 045416 016701 134474      MOV      L$DLY,R1     : *,SSTMP1
13011 045422 001403      BEQ     9$           :
13012 045424 005016      8$:      CLR      (SP)        : SSTMP
13013 045426 005301      DEC     R1           : SSTMP1
13014 045430 001375      BNE     8$          :
13015 045432 005300      9$:      DEC     R0           : SSTMP2
13016 045434 000767      BR       7$          :
13017 045436 132777 000004 145570      10$:     BITB     #4,@ML.REG+60 :
13018 045444 001024      BNE     11$          :

```

6338
6365
6366
6367
6369
6371
6375
6371
6378
6371
6381
6384
6386

```

13020      ;ML4
13021      ;
13022
13023 045446 104455      TRAP      55      ;
13024 045450 000105      .WORD     105      ;
13025 045452 010464      .WORD     ASYNC      ;
13026 045454 024052      .WORD     DUMPER      ;
13027 045456 012746 006522  MOV      #WORD.21,-(SP) ;
13028 045462 012746 006416  MOV      #WORD.11,-(SP) ;
13029 045466 012746 007366  MOV      #PHR.1,-(SP)   ;
13030 045472 012746 006522  MOV      #WORD.21,-(SP) ;
13031 045476 012746 006046  MOV      #FOR.FMT,-(SP) ;
13032 045502 012746 000005  MOV      #5,-(SP)      ;
13033 045506 010600      MOV      SP,R0        ; SP,*
13034 045510 104414      TRAP      14
13035 045512 062706 000014  ADD      #14,SP        ;
13036 045516 032777 040000 145430 11$: BIT      #40000,@ML.REG ;
13037 045524 001024      BNE      12$          ;
13038 045526 104455      TRAP      55          ;
13039 045530 000165      .WORD     165          ;
13040 045532 010526      .WORD     SYNC          ;
13041 045534 024052      .WORD     DUMPER          ;
13042 045536 012746 006522  MOV      #WORD.21,-(SP) ;
13043 045542 012746 006416  MOV      #WORD.11,-(SP) ;
13044 045546 012746 007366  MOV      #PHR.1,-(SP)   ;
13045 045552 012746 006514  MOV      #WORD.20,-(SP) ;
13046 045556 012746 006046  MOV      #FOR.FMT,-(SP) ;
13047 045562 012746 000005  MOV      #5,-(SP)      ;
13048 045566 010600      MOV      SP,R0        ; SP,*
13049 045570 104414      TRAP      14
13050 045572 062706 000014  ADD      #14,SP        ;
13051 045576 104467      12$: TRAP      67      ;
13052 045600 006000      ROR      R0            ;
13053 045602 103641      BLO      1$            ;
13054 045604 005202      INC      R2            ; CNT
13055 045606 020227 000002  CMP      R2,#2        ; CNT,*
13056 045612 003635      BLE      1$            ;
13057 045614 005726      TST      (SP)+        ;
13058 045616 000207      RTS      PC            ;
13059
13060
13061
13066
13067
13071

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

6389

6390

6388

6393

6396

6397

6395

6398

6365

6338

; Routine Size: 105 words
; Maximum stack depth per invocation: 10 words

13079 045620
13080 045620 004767 177452
13081 045624 104466
13082 045626 006000
13083 045630 103773
13084 045632 000207

T23::
1\$: JSR PC,\$T23
TRAP 66
ROR R0
BLO 1\$
RTS PC

6401

13085
13086 : Routine Size: 6 words
13087 : Maximum stack depth per invocation: 0 words
13092
13093
13094 : 6404 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (74)

```

13096 :ML4
13097 :
13098 :
13099 :      6405 :
13100 :      6406 :
13101 :      6407 : BGNTST;
13102 :      6408 :
13103 :      6409 : !++
13104 :      6410 : TEST NUMBER: TST 24
13105 :      6411 :
13106 :      6412 : TEST NAME : initial PROM TEST
13107 :      6413 :
13108 :      6414 : TEST DESCRIPTION:
13109 :      6415 :
13110 :      6416 : TEST THE MEMORY ARRAYS' PROM
13111 :      6417 : TIMING AND CONTROL LOGIC FOR
13112 :      6418 : INITIAL PROM READS BY:
13113 :      6419 :
13114 :      6420 : 1. READING 14 PROM LOCATIONS
13115 :      6421 : AND TESTING FOR:
13116 :      6422 :
13117 :      6423 : A. CHECK SUM ERRORS AT
13118 :      6424 : EACH ROW COLUMN ADRS
13119 :      6425 :
13120 :      6426 : B. SUM OF EACH PROM BIT SET
13121 :      6427 : <9,0> GTR 14.
13122 :      6428 :
13123 :      6429 :
13124 :      6430 :
13125 :      6431 : local
13126 :      6432 : OFF_SET_CNT : vector [10, byte],
13127 :      6433 : ROW_ORED_COL : bitvector [16],
13128 :      6434 : R_C_SAV : bitvector [16],
13129 :      6435 : PROM_ADRS,
13130 :      6436 : CHK_SUM,
13131 :      6437 : CHK_SUM_ERR,
13132 :      6438 : BAD_NIB_CNT,
13133 :      6439 : CNT_14_BAD,
13134 :      6440 : DODD_FLG,
13135 :      6441 : ERR_FLG;
13136 :      6442 :
13137 :      6443 : CLR MBUS;
13138 :      6444 : DODD_FLG = ZERO;
13139 :      6445 : PROM_DIS = ONE;
13140 :      6446 : CHK_SUM_ERR = ZEROES;
13141 :      6447 :
13142 :      6448 : incr CNT from 0 to 9 do
13143 :      6449 : OFF_SET_CNT [.CNT] = ZEROES;
13144 :      6450 :
13145 :      6451 : incr ADRS_CNT from 0 to 14 do
13146 :      6452 : begin
13147 :      6453 : ROW_ORED_COL = ZEROES;
13148 :      6454 : PROM_ADRS = .ADRS_CNT;
13149 :      6455 :
13150 :      6456 : incr TWICE from 0 to 1 do

```

```

!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
!GET COPY OF ADRS_CNT
```

```
!READ ROW AND COL DATA FOR THIS ARRAY WORD
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (74)

```

13152 :ML4
13153 :
13154 :
13155 :      6457      begin
13156 :      6458      BAD_NIB_CNT = ZEROES;
13157 :      6459      ERR_FLG = ZERO;
13158 :      6460      MLPA = .PROM_ADRS;          !LOADING MLPA INITIATES PROM READ
13159 :      6461      DELAY (ONE_US);
13160 :      6462      R_C_SAV = .MLPD;          !GET THE ROW OR COL DATA
13161 :      6463
13162 :      6464      incr BIT_CNT from 0 to 9 do          !COUNT NUMBER BITS SET IN <9:0>
13163 :      6465
13164 :      6466          if .R_C_SAV [.BIT_CNT] IS_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;
13165 :      6467
13166 :      6468      CHK_SUM = .R_C_SAV<10, 3>;          !GET THE CHECK SUM BITS
13167 :      6469
13168 :      6470      if .R_C_SAV [13] IS_SET then CHK_SUM = .CHK_SUM + 1;      !ADD IN BIT 13
13169 :      6471
13170 :      6472      if .R_C_SAV [14] IS_SET then CHK_SUM = .CHK_SUM + 1;      !ADD IN BIT 14
13171 :      6473
13172 :      6474      if .R_C_SAV [15] IS_SET then CHK_SUM = .CHK_SUM + 1;      !ADD IN BIT 15
13173 :      6475
13174 :      6476      if .R_C_SAV [15] IS_SET          !IS BIT 15 SET
13175 :      6477      then
13176 :      6478          begin
13177 :      6479
13178 :      6480          if .BAD_NIB_CNT lss .CHK_SUM then ERR_FLG = ONE;      !SET ERROR FLG IF CHECK SUM ERROR
13179 :      6481
13180 :      6482          end
13181 :      6483      else
13182 :      6484          begin
13183 :      6485
13184 :      6486          if .BAD_NIB_CNT neq .CHK_SUM then ERR_FLG = ONE;      !SET ERROR FLG IF CHECK SUM ERROR
13185 :      6487
13186 :      6488          end;
13187 :      6489
13188 :      6490      if .ERR_FLG IS_SET          !WAS THERE A CHECK SUM ERROR
13189 :      6491      then
13190 :      6492          begin          !REPORT INTERMEDIATE ERROR IF YES
13191 :      6493          ERRDF (70, INTER, DUMPER);
13192 :      6494          PRINTB (SIX_FMT, FNC_21, WRD_10, WRD_12, WRD_45, WRD_35, FNC_6);
13193 :      6495          PRINTB (FMT_10, .CHK_SUM, .R_C_SAV);
13194 :      6496          DODU_FLG = ONE;
13195 :      6497          end;
13196 :      6498
13197 :      6499      ROW_ORED_COL = (.ROW_ORED_COL) or (.R_C_SAV);      !OR ROW AND COLUMN DATA
13198 :      6500      PROM_ADRS = (.PROM_ADRS) or (%o'2000');      !GET COLUMN DATA
13199 :      6501      end;
13200 :      6502
13201 :      6503      incr index from 0 to 9 do          !GET TOTAL OFF SET COUNTS FROM THE 15 ARRAY WORDS
13202 :      6504
13203 :      6505          if .ROW_ORED_COL [.index] IS_SET then OFF_SET_CNT [.index] = .OFF_SET_CNT [.index] + 1;
13204 :      6506
13205 :      6507      end;
13206 :      6508

```


Line	Address	OpCode	Operand	Label	Comment	Time	Page
13263					:ML4	22-Dec-1980 09:24:31	TOPS
13264					:	22-Dec-1980 09:21:22	PA:C
13265							
13266	045700	152777	000040	145366	BISB #40,@ML.REG+120		6445
13267	045706	005002			CLR R2	: CHK.SUM.ERR	6446
13268	045710	012701	000016	1\$:	MOV #16,R1	: OFF.SET.CNT,*	6449
13269	045714	060601			ADD SP,R1	: CNT,*	
13270	045716	060201			ADD R2,R1		
13271	045720	105011			CLRB (R1)	: CNT	6448
13272	045722	005202			INC R2	: CNT,*	
13273	045724	020227	000011		CMP R2,#11		
13274	045730	003767			BLE 1\$		
13275	045732	005005			CLR R5	: ADRS.CNT	6451
13276	045734	005066	000010	2\$:	CLR 10(SP)	: ROW.ORED.COL	6453
13277	045740	010566	000006		MOV R5,6(SP)	: ADRS.CNT,PROM.ADRS	6454
13278	045744	005004			CLR R4	: TWICE	6456
13279	045746	005066	000004	3\$:	CLR 4(SP)	: BAD.NIB.CNT	6458
13280	045752	005016			CLR (SP)	: ERR.FLG	6459
13281	045754	016677	000006	145272	MOV 6(SP),@ML.REG+100	: PROM.ADRS,*	6460
13282	045762	012701	000001		MOV #1,R1	: *,SSTMP2	6461
13283	045766	001411			BEQ 7\$		
13284	045770	016702	134122	4\$:	MOV LSDLY,R2	: *,SSTMP1	
13285	045774	001404			BEQ 6\$		
13286	045776	005066	000014	5\$:	CLR 14(SP)	: SSTMP	
13287	046002	005302			DEC R2	: SSTMP1	
13288	046004	001374			BNE 5\$		
13289	046006	005301		6\$:	DEC R1	: SSTMP2	
13290	046010	000766			BR 4\$		
13291	046012	017766	145366	000012	7\$:	: *,R.C.SAV	6462
13292	046020	005002			CLR R2	: BIT.CNT	6464
13293	046022	010201			8\$:	: BIT.CNT,*	6466
13294	046024	006201			MOV R2,R1		
13295	046026	006201			ASR R1		
13296	046028	006201			ASR R1		
13297	046030	012700	000012		ASR R1		
13298	046032	006201			MOV #12,R0		
13299	046034	006201			ADD SP,R0	: R.C.SAV,*	
13300	046036	006201			ADD R0,R1		
13301	046038	010146			MOV R1,-(SP)		
13302	046040	010246			MOV R2,-(SP)	: BIT.CNT,*	
13303	046042	042716	177770		BIC #177770,(SP)		
13304	046044	012746	000001		MOV #1,-(SP)		
13305	046046	006201			CLR -(SP)		
13306	046048	004767	135054		JSR PC,BLSGT2		
13307	046050	006201			ADD #10,SP		
13308	046052	005300			DEC R0		
13309	046054	001002			BNE 9\$		
13310	046056	005266	000004		INC 4(SP)	: BAD.NIB.CNT	
13311	046100	005202			INC R2	: BIT.CNT	6464
13312	046102	020227	000011	9\$:	CMP R2,#11	: BIT.CNT,*	
13313	046106	003745			BLE 8\$		
13314	046110	016603	000012		MOV 12(SP),R3	: R.C.SAV,CHK.SUM	6468
13315	046114	006203			ASR R3	: CHK.SUM	
13316	046116	006203			ASR R3	: CHK.SUM	
13317	046120	000303			SWAB R3	: CHK.SUM	
13318	046122	042703	177770		BIC #177770,R3	: *,CHK.SUM	

Address	Hex	Hex	Hex	Hex	Label	Instruction	Comments	Address
13319					:ML4			
13320					:			
13321								
13322	046126	132766	000040	000013		BITB #40,13(SP)	: *,R. .AV+1	6470
13323	046134	001401				BEQ 10\$		
13324	046136	005203				INC R3	: CHK.SUM	
13325	046140	132766	000100	000013	10\$:	BITB #100,13(SP)	: *,R.C.SAV+1	6472
13326	046146	001401				BEQ 11\$		
13327	046150	005203				INC R3	: CHK.SUM	
13328	046152	005002			11\$:	CLR R2		6474
13329	046154	105766	000013			TSTB 13(SP)	: R.C.SAV+1	
13330	046160	100002				BPL 12\$		
13331	046162	005203				INC R2		
13332	046164	005203				INC R3	: CHK.SUM	
13333	046166	006002			12\$:	ROR R2		6476
13334	046170	103004				BCC 13\$		
13335	046172	026303	000004			CMP 4(SP),R3	: BAD.NIB.CNT,CHK.SUM	6480
13336	046176	002006				BGE 15\$		
13337	046200	000403				BR 14\$		
13338	046202	026603	000004		13\$:	CMP 4(SP),R3	: BAD.NIB.CNT,CHK.SUM	6486
13339	046206	001402				BEQ 15\$		
13340	046210	012716	000001		14\$:	MOV #1,(SP)	: *,ERR.FLG	
13341	046214	021627	000001		15\$:	CMP (SP),#1	: ERR.FLG,*	6490
13342	046220	001044				BNE 16\$		
13343	046222	104455				TRAP 55		6493
13344	046224	000106				.WORD 106		
13345	046226	010672				.WORD INTER		
13346	046230	024052				.WORD DUMPER		
13347	046232	012746	010004			MOV #FNC.6,-(SP)		6494
13348	046236	012746	006666			MOV #WORD.35,-(SP)		
13349	046242	012746	006772			MOV #WORD.45,-(SP)		
13350	046246	012746	006426			MOV #WORD.12,-(SP)		
13351	046252	012746	006406			MOV #WORD.10,-(SP)		
13352	046256	012746	010216			MOV #FNC.21,-(SP)		
13353	046262	012746	006100			MOV #SIX.FMT,-(SP)		
13354	046266	012746	000007			MOV #7,-(SP)		
13355	046272	010600				MOV SP,R0	: SP,*	
13356	046274	104414				TRAP 14		
13357	046276	016616	000032			MOV 32(SP),(SP)	: R.C.SAV,*	6495
13358	046282	010346				MOV R3,-(SP)	: CHK.SUM,*	
13359	046304	012746	004676			MOV #FMT.10,-(SP)		
13360	046310	012746	000003			MOV #3,-(SP)		
13361	046314	010600				MOV SP,R0	: SP,*	
13362	046316	104414				TRAP 14		
13363	046320	012766	000001	000030		MOV #1,30(SP)	: *,DODU.FLG	6496
13364	046326	062706	000026			ADD #26,SP		6492
13365	046332	056666	000012	000010	16\$:	BIS 12(SP),10(SP)	: R.C.SAV,ROW.ORED.COL	6499
13366	046340	052766	002000	003006		BIS #2000,6(SP)	: *,PROM.ADRS	6500
13367	046346	005204				INC R4	: TWICE	6456
13368	046350	020427	000001			CMP R4,#1	: TWICE,*	
13369	046354	003002				BGT 17\$		
13370	046356	000167	177364			JMP 3\$		
13371	046362	005002			17\$:	CLR R2	: INDEX	6503
13372	046364	010201			18\$:	MOV R2,R1	: INDEX.*	6505
13373	046366	006201				ASR R1		

```

13375      :ML4
13376      :
13377
13378 046370 006201      ASR      R1
13379 046372 006201      ASR      R1
13380 046374 012700 000010  MOV     #10,R0
13381 046400 060600      ADD     SP,R0      ; ROW.ORED.COL,*
13382 046402 060001      ADD     R0,R1
13383 046404 010146      MOV     R1,-(SP)
13384 046406 010246      MOV     R2,-(SP)      ; INDEX,*
13385 046410 042716 177770  BIC     #177770,(SP)
13386 046414 012746 000001  MOV     #1,-(SP)
13387 046420 005046      CLR     -(SP)
13388 046422 004767 134512  JSR     PC,BLSGT2
13389 046426 062706 000010  ADD     #10,SP
13390 046432 005300      DEC     R0
13391 046434 001005      BNE     19$
13392 046436 012701 000016  MOV     #16,R1
13393 046442 060601      ADD     SP,R1      ; OFF.SET.CNT,*
13394 046444 060201      ADD     R2,R1      ; INDEX,*
13395 046446 105211      INCB   (R1)
13396 046450 005202      INC     R2      ; INDEX      6503
13397 046452 020227 000011  CMP     R2,#11      ; INDEX,*
13398 046456 003742      BLE     18$
13399 046460 005205      INC     R5      ; ADRS.CNT      6451
13400 046462 020527 000016  CMP     R5,#16      ; ADRS.CNT,*
13401 046466 000002      BGT     20$
13402 046470 000167 177240  JMP     2$
13403 046474 005000      CLR     R0      ; CNT.14.BAD      6509
13404 046476 005001      CLR     R1      ; CNT      6511
13405 046500 012702 000016  MOV     #16,R2      ; CNT      6513
13406 046504 060502      ADD     SP,R2
13407 046506 060102      ADD     R1,R2      ; OFF.SET.CNT,*
13408 046510 121227 000016  CMPB   (R2),#16      ; CNT,*
13409 046514 103401      BLO.   22$
13410 046516 005200      INC     R0      ; CNT.14.BAD
13411 046520 005201      INC     R1      ; CNT      6511
13412 046522 020127 000011  CMP     R1,#11      ; CNT,*
13413 046526 003764      BLE     21$
13414 046530 005700      TST     R0      ; CNT.14.BAD      6515
13415 046532 001505      BEQ     27$
13416 046534 104455      TRAP   $S
13417 046536 000107      .WORD  107
13418 046540 010570      .WORD  ARR.DAT
13419 046542 024052      .WORD  DUMPER
13420 046544 020027 000012  CMP     R0,#12      ; CNT.14.BAD,*      6520
13421 046550 001024      BNE     23$
13422 046552 012746 006726  MOV     #WORD.40,-(SP)
13423 046556 012746 006746  MOV     #WORD.42,-(SP)
13424 046562 012746 007054  MOV     #WORD.51,-(SP)
13425 046566 012746 007004  MOV     #WORD.46,-(SP)
13426 046572 012746 007104  MOV     #WORD.54,-(SP)
13427 046576 012746 007114  MOV     #WORD.55,-(SP)
13428 046602 012746 006100  MOV     #SIX.FMT,-(SP)
13429 046606 012746 000007  MOV     #7,-(SP)

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

```

13431      :ML4
13432      :
13433      :
13434 046612 010600      MOV      SP,R0      ; SP,*
13435 046614 104414      TRAP     14
13436 046616 022626      CMP      (SP)+,(SP)+
13437 046620 000445      BR       26$
13438 046622 012746 006726 23$:      MOV      #WORD.40,-(SP)
13439 046626 012746 006746      MOV      #WORD.42,-(SP)
13440 046632 012746 007016      MOV      #WORD.47,-(SP)
13441 046636 012746 007004      MOV      #WORD.46,-(SP)
13442 046642 012746 006046      MOV      #FOR.FMT,-(SP)
13443 046646 012746 000005      MOV      #5,-(SP)
13444 046652 010600      MOV      SP,R0      ; SP,*
13445 046654 104414      TRAP     14
13446 046656 005002      CLR      R2          ; CNT
13447 046660 012701 000032 24$:      MOV      #32,R1
13448 046664 060601      ADD      SP,R1      ; OFF.SET.CNT,*
13449 046666 060201      ADD      R2,R1      ; CNT,*
13450 046670 121127 000016      CMPB    (R1),#16
13451 046674 103413      BLO     25$
13452 046676 005046      CLR     -(SP)
13453 046700 111116      MOVB    (R1),(SP)
13454 046702 010246      MOV     R2,-(SP)    ; CNT,*
13455 046704 012746 005034      MOV     #FMT.13,-(SP)
13456 046710 012746 000003      MOV     #3,-(SP)
13457 046714 010600      MOV     SP,R0      ; SP,*
13458 046716 104414      TRAP     14
13459 046720 062706 000010      ADD     #10,SP
13460 046724 005202 25$:      INC     R2          ; CNT
13461 046726 020227 000011      CMP     R2,#11     ; CNT,*
13462 046732 003752      BLE     24$
13463 046734 012766 000001 000016 26$:      MOV     #1,16(SP)  ; *,DODU.FLG
13464 046742 062706 000014      ADD     #14,SP
13465 046746 026627 000002 000001 27$:      CMP     2(SP),#1  ; DODU.FLG,*
13466 046754 001004      BNE     28$
13467 046756 016700 144616      MOV     ML.LUN,R0
13468 046762 104451      TRAP     51
13469 046764 104444      TRAP     44
13470 046766 062706 000030 28$:      ADD     #30,SP
13471 046772 000207      RTS     PC

```

; Routine Size: 304 words
; Maximum stack depth per invocation: 29 words

13472
13473
13474
13479
13480
13484

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

13486
13487
13488
13492 046774
13493 046774 004767 176634
13494 047000 104466
13495 047002 006000
13496 047004 103773
13497 047006 000207
13498
13499
13500
13505
13506
13507 :

:ML4
:

T24::
1\$:

JSR PC,\$T24
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

6545 !<BLF/PAGE>

6542

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (75)

```

13509 :ML4
13510 :
13511 :
13512 :      6546  |
13513 :      6547  |
13514 :      6548  | BGNTST:
13515 :      6549  |
13516 :      6550  | !++
13517 :      6551  | TEST NUMBER: TST 25
13518 :      6552  |
13519 :      6553  | TEST NAME:  PROM OR FUNCTION TEST
13520 :      6554  |
13521 :      6555  | TEST DESCRIPTION:
13522 :      6556  |
13523 :      6557  | TEST THE HARDWARE ORING OF
13524 :      6558  | THE MEMORY ARRAYS' PROM
13525 :      6559  | ROW COLUMN DATA BY:
13526 :      6560  |
13527 :      6561  | 1.  READING AND STORING 128
13528 :      6562  |     HARDWARE ORED ROW COLUMN
13529 :      6563  |     DATA.
13530 :      6564  |
13531 :      6565  | 2.  THEN IN PROM DISABLE MODE
13532 :      6566  |     AND VIA SOFTWARE CONTROL,
13533 :      6567  |     READ AND OR PROM ROW
13534 :      6568  |     COLUMN DATA AND COMPARE
13535 :      6569  |     AGAINST THE RESPECTIVE
13536 :      6570  |     STORED HARDWARE ORED DATA.
13537 :      6571  |
13538 :      6572  |
13539 :      6573  |
13540 :      6574  | local
13541 :      6575  |   R_BITS,
13542 :      6576  |   C_BITS,
13543 :      6577  |   SW_ORED,
13544 :      6578  |   HW_SAVE,
13545 :      6579  |   DODU_FLG;
13546 :      6580  |
13547 :      6581  | CLR MBUS;
13548 :      6582  | DODU_FLG = ZERO;
13549 :      6583  | DAT_DM = ONE;
13550 :      6584  | FIRST_BLK_XFER ();
13551 :      6585  | ML_FUNC = write;
13552 :      6586  |
13553 :      6587  | incr PROM_ADRS from 0 to 127 do
13554 :      6588  |   begin
13555 :      6589  |     DELAY (ONE US);
13556 :      6590  |     HW OR TBL [.PROM_ADRS] = .MLPD;
13557 :      6591  |     DAT_CLK = ONE;
13558 :      6592  |   end;
13559 :      6593  |
13560 :      6594  | CLR MBUS;
13561 :      6595  | PROM_DIS = ONE;
13562 :      6596  |
13563 :      6597  | incr PROM_ADRS from 0 to 127 do

```

```

!PROM ROW DATA
!PROM COL DATA
!SOFTWARE CALCULATED FROM 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

!SET PROM DISABLE MODE

!CALCULATE 128 SW ORED DATA & COMPARE TO HW TABLE

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (75)

```

13565 :ML4
13566 :
13567 :
13568 :      6598      begin
13569 :      6599      MLPA = .PROM ADRS;           !LOADING MLPA INITIATES A PROM READ
13570 :      6600      DELAY (ONE US);
13571 :      6601      R BITS = .ALPD;             !SAVE ROW DATA
13572 :      6602      MLPA = .PROM ADRS or %o'2000'; !ENABLE COLUMN DATA ADRS
13573 :      6603      DELAY (ONE US);
13574 :      6604      C BITS = .ALPD;             !SAVE COL DATA
13575 :      6605      SW_ORED = .R BITS or .C BITS; !CALCULATE SOFTWARE ORED
13576 :      6606      HW_SAVE = .HW_OR_TBL [.PROM_ADRS]; !GET RESPECTIVE HARDWARE ORED
13577 :      6607
13578 :      6608      if (.HW_SAVE<0, 9>) nea (.SW_ORED<0, 9>) !COMPARE SW & HW ORED
13579 :      6609      then
13580 :      6610          begin                     !IF NEG THEN ERROR
13581 :      6611          ERRDF (76, ARR DAT, DUMPER);
13582 :      6612          PRINTB (FOR_FMT, WRD 35, WRD 36, WRD 19, PHR 4);
13583 :      6613          PRINTB (FMT_2, .SW_ORED<0, 9>, .HW_SAVE<0, 9>, (.SW_ORED<0, 9> xor .HW_SAVE<0, 9>));
13584 :      6614          DODU_FLG = ONE;
13585 :      6615          end;
13586 :      6616
13587 :      6617      end;
13588 :      6618
13589 :      6619      if .DODU_FLG IS_SET           !DROP THIS UNIT IF DODU FLG IS_SET
13590 :      6620      then
13591 :      6621          begin
13592 :      6622          DODU (.ML_LUN);
13593 :      6623          DOCLN;
13594 :      6624          end;
13595 :      6625
13596 :      6626      ENDTST;

```

```

13604 047010 004167 135102      $T25:  ISR      R1,$SAVE5           ;           6544
13605 047014 162706 000010      SUB      #10,SP           ;           6579
13606 047020 152777 000040 144166  BISB    #40,@ML.REG+40    ;
13607 047026 016705 144550      MOV     ML,DUT,R5
13608 047032 042705 177770      BIC    #177770,R5
13609 047036 142777 000007 144150  BICB   #7,@ML.REG+40
13610 047044 150577 144144      BISB   R5,@ML.REG+40
13611 047050 005066 000004      CLR    4(SP)              ; DODU.FLG 6582
13612 047054 152777 000010 144212  BISB   #10,@ML.REG+120   ;           6583
13613 047062 004767 146752      JSR    PC,FIRST.BLK.XFER ;           6584
13614 047066 142777 000077 144060  BICB   #77,@ML.REG      ;           6585
13615 047074 152777 000061 144052  BISB   #61,@ML.REG
13616 047102 005002      CLR    R2                 ; PROM.ADRS 6587
13617 047104 012701 000001      $S:    MOV    #1,R1        ; *,$TMP2   6589
13618 047110 001411      2$:    BEQ    5$

```

```

13620          :ML4
13621          :
13622          :
13623 047112 016703 133000          MOV      LSDLY,R3          : *,SSTMP1
13624 047116 001404          BEQ      4$
13625 047120 005066 000006          3$: CLR      6(SP)          : SSTMP
13626 047124 005303          DEC      R3              : SSTMP1
13627 047126 001374          BNE     3$
13628 047130 005301          4$: DEC      R1              : SSTMP2
13629 047132 000766          BR       2$
13630 047134 010203          5$: MOV      R2,R3          : PROM.ADRS,*
13631 047136 006303          ASL      R3
13632 047140 017763 144240 011160 MOV      @ML.REG+230,HJ.OR.TBL(R3)
13633 047146 152777 000020 144120 BISB     #20,@ML.REG+120
13634 047154 005202          INC      R2              : PROM.ADRS
13635 047156 020227 000177          CMP      R2,#177         : PROM.ADRS,*
13636 047162 003750          BLE     1$
13637 047164 152777 000040 144022 BISB     #40,@ML.REG+40
13638 047172 016705 144404          MOV      ML,DUT,R5
13639 047176 042705 177770          BIC     #177770,R5
13640 047202 142777 000007 144004 BICB     #7,@ML.REG+40
13641 047210 150577 144000          BISB     R5,@ML.REG+40
13642 047214 152777 000040 144052 BISB     #40,@ML.REG+120
13643 047222 005001          CLR      R1              : PROM.ADRS
13644 047224 010177 144024          6$: MOV      R1,@ML.REG+100 : PROM.ADRS,*
13645 047230 012702 000001          MOV      #1,R2          : *,SSTMP2
13646 047234 001411          7$: BEQ      10$
13647 047236 016703 132654          MOV      LSDLY,R3          : *,SSTMP1
13648 047242 001404          BEQ      9$
13649 047244 005066 000006          8$: CLR      6(SP)          : SSTMP
13650 047250 005303          DEC      R3              : SSTMP1
13651 047252 001374          BNE     8$
13652 047254 005302          9$: DEC      R2              : SSTMP2
13653 047256 000766          BR       7$
13654 047260 017766 144120 000002 10$: MOV      @ML.REG+230,2(SP) : *,R.BITS
13655 047266 010103          MOV      R1,R3          : PROM.ADRS,*
13656 047270 052703 002000          BIS      #2000,R3
13657 047274 010377 143754          MOV      R3,@ML.REG+100
13658 047300 012702 000001          MOV      #1,R2          : *,SSTMP2
13659 047304 001411          11$: BEQ      14$
13660 047306 016703 132604          MOV      LSDLY,R3          : *,SSTMP1
13661 047312 001404          BEQ      13$
13662 047314 005066 000006          12$: CLR      6(SP)          : SSTMP
13663 047320 005303          DEC      R3              : SSTMP1
13664 047322 001374          BNE     12$
13665 047324 005302          13$: DEC      R2              : SSTMP2
13666 047326 000766          BR       11$
13667 047330 017716 144050          14$: MOV      @ML.REG+230,(SP) : *,C.BITS
13668 047334 016605 000002          MOV      2(SP),R5        : R.BITS,SW.ORED
13669 047340 051605          BIS      (SP),R5         : C.BITS,SW.ORED
13670 047342 010103          MOV      R1,R3          : PROM.ADRS,*
13671 047344 006303          ASL      R3
13672 047346 016304 011160          MOV      HJ.OR.TBL(R3),R4 : *,HJ.SAVE
13673 047352 010502          MOV      R5,R2          : SW.ORED,*
13674 047354 042702 177000          BIC     #177000,R2

```

6590

6591

6587

6592

6595

6597

6599

6600

6601

6602

6603

6604

6605

6606

6608


```

13676      :ML4
13677      :
13678
13679 047360 010403      MOV    R4,R3           ; HW.SAVE,*
13680 047362 042703 177000 BIC    #177000,R3
13681 047366 020302      CMP    R3,R2
13682 047370 001451      BEQ   15$
13683 047372 104455      TRAP  55
13684 047374 000114      .WORD 114             ;
13685 047376 010570      .WORD ARR.DAT
13686 047400 024052      .WORD DUMPER
13687 047402 012746 007454 MOV    #PHR.4,-(SP)    ;
13688 047406 012746 006506 MOV    #WRD.19,-(SP)  ;
13689 047412 012746 006674 MOV    #WRD.36,-(SP)  ;
13690 047416 012746 006666 MOV    #WRD.35,-(SP)  ;
13691 047422 012746 006046 MOV    #FOR.FMT,-(SP) ;
13692 047426 012746 000005 MOV    #5,-(SP)
13693 047432 010600      MOV    SP,R0          ; SP,*
13694 047434 104414      TRAP  14
13695 047436 010403      MOV    R4,R3           ; HW.SAVE,*
13696 047440 010516      MOV    R5,(SP)         ; SW.ORED,*
13697 047442 010302      MOV    R3,R2
13698 047444 040502      BIC   R5,R2
13699 047446 040316      BIC   R3,(SP)
13700 047450 050216      BIS   R2,(SP)
13701 047452 010446      MOV    R4,-(SP)        ; HW.SAVE,*
13702 047454 042716 177000 BIC    #177000,(SP)
13703 047460 010546      MOV    R5,-(SP)        ; SW.ORED,*
13704 047462 042716 177000 BIC    #177000,(SP)
13705 047466 012746 004266 MOV    #FMT.2,-(SP)
13706 047472 012746 000004 MOV    #4,-(SP)
13707 047476 010600      MOV    SP,R0          ; SP,*
13708 047500 104414      TRAP  14
13709 047502 012766 000001 000030 MOV    #1,30(SP)      ; *,DODU.FLG
13710 047510 062706 000024      ADD   #24,SP
13711 047514 005201      15$: INC   R1             ;
13712 047516 020127 000177      CMP   R1,#177         ; PROM.ADRS
13713 047522 003640      BLE  68               ; PROM.ADRS,*
13714 047524 026627 000004 000001 CMP    4(SP),#1       ; DODU.FLG,*
13715 047532 001004      BNE  168
13716 047534 016700 144040 MOV    ML.LUN,R0      ;
13717 047540 104451      TRAP  51
13718 047542 104444      TRAP  44
13719 047544 062706 000010 16$: ADD   #10,SP
13720 047550 000207      RTS   PC
13721
13722      ; Routine Size: 177 words
13723      ; Maximum stack depth per invocation: 20 words
13728
13729

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

6611

6612

6613

6614

6610

6597

6619

6622

6544

13737
13741 047552
13742 047552 004767 177232
13743 047556 104466
13744 047560 006000
13745 047562 103773
13746 047564 000207
13747
13748
13749
13754
13755
13756 :

T25::
1\$: JSR PC,\$T25 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

6624

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

6627 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (76)

13758 :ML4
13759 :
13760 :
13761 :
13762 :
13763 :
13764 :
13765 :
13766 :
13767 :
13768 :
13769 :
13770 :
13771 :
13772 :
13773 :
13774 :
13775 :
13776 :
13777 :
13778 :
13779 :
13780 :
13781 :
13782 :
13783 :
13784 :
13785 :
13786 :
13787 :
13788 :
13789 :
13790 :
13791 :
13792 :
13793 :
13794 :
13795 :
13796 :
13797 :
13798 :
13799 :
13800 :
13801 :
13802 :
13803 :
13804 :
13805 :
13806 :
13807 :
13808 :
13809 :
13810 :
13811 :
13812 :

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
6675
6676
6677
6678
6679

BGNTST;

!++

TEST NUMBER: TST 26

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.

!--

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;
BGNSDB;
CLR_MBUS;
ERR_FLG = ZERO;
LOW_CNT = ZEROES;
HIGH_CNT = ZEROES;
GTR_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

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (76)

```

13870 :ML4
13871 :
13872 :
13873 :           6732      ERRDF (73, ARR_DAT, DUMPER);
13874 :           6733      PRINTB (SIX_FMT, WRD_34, PHR_5, WRD_32, WRD_6, WRD_33, WRD_24);
13875 :           6734      ERR_FLG = ONE;
13876 :           6735      end;
13877 :           6736
13878 :           6737      end
13879 :           6738
13880 :           6739      end
13881 :           6740      else
13882 :           6741      begin
13883 :           6742
13884 :           6743      if .GTR_FLG IS_SET
13885 :           6744      then
13886 :           6745      begin
13887 :           6746      TEMP = .PROM_DATA;
13888 :           6747      TEMP = .TEMP and %o'162000';
13889 :           6748
13890 :           6749      if .TEMP neq %o'162000'
13891 :           6750      then
13892 :           6751      begin
13893 :           6752      ERRDF (74, ARR_DAT, DUMPER);
13894 :           6753      PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
13895 :           6754      ERR_FLG = ONE;
13896 :           6755      end
13897 :           6756      else
13898 :           6757      begin
13899 :           6758
13900 :           6759      if .LOW_CNT lss .HIGH_CNT
13901 :           6760      then
13902 :           6761      begin
13903 :           6762      ERRDF (75, ARR_DAT, DUMPER);
13904 :           6763      PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
13905 :           6764      ERR_FLG = ONE;
13906 :           6765      end;
13907 :           6766
13908 :           6767      end
13909 :           6768
13910 :           6769      end
13911 :           6770      else
13912 :           6771      begin
13913 :           6772
13914 :           6773      if .LOW_CNT neq .HIGH_CNT
13915 :           6774      then
13916 :           6775      begin
13917 :           6776      ERRDF (108, ARR_DAT, DUMPER);
13918 :           6777      PRINTB (SIX_FMT, WRD_34, PHR_1, WRD_32, WRD_5, WRD_33, WRD_24);
13919 :           6778      ERR_FLG = ONE;
13920 :           6779      end
13921 :           6780
13922 :           6781      end
13923 :           6782
13924 :           6783      end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (76)

```

13926 :ML4
13927 :
13928 :
13929 :      6784
13930 :      6785      if .ERR_FLG IS_SET
13931 :      6786      then
13932 :      6787          begin
13933 :      6788          PRINTB (FMT 7, .PROM_DATA);
13934 :      6789          DODU_FLG = ONE;
13935 :      6790          end;
13936 :      6791
13937 :      6792      ENDSUB;
13938 :      6793      end
13939 :      6794      until .PROM_DATA eql %o'177777';
13940 :      6795
13941 :      6796      if .DODU_FLG IS_SET
13942 :      6797      then
13943 :      6798          begin
13944 :      6799          DODU (.ML_LUN);
13945 :      6800          DOCLN;
13946 :      6801          end;
13947 :      6802
13948 :      6803      ENDTST;
13952 :

```

```

!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

```

13956	047566	004167	134324	ST26:	JSR	R1, \$SAVE5	:	6626
13957	047572	162706	000012		SUB	#12, SP	:	
13958	047576	005066	000006		CLR	6(SP)	:	DODU.FLG 6668
13959	047602	012766	177777	000010	MOV	#-1, 10(SP)	:	*.PROM.DATA 6669
13960	047610	005266	000010	1S:	INC	10(SP)	:	PROM.DATA 6673
13961	047614	104402		2S:	TRAP	2	:	
13962	047616	152777	000040	143370	BISB	#40, @ML.REG+40	:	6674
13963	047624	016705	143752		MOV	ML.DUT, R5	:	
13964	047630	042705	177770		BIC	#177770, R5	:	
13965	047634	142777	000007	143352	BICB	#7, @ML.REG+40	:	
13966	047642	150577	143346		BISB	R5, @ML.REG+40	:	
13967	047646	005001			CLR	R1	:	ERR.FLG 6676
13968	047650	005066	000002		CLR	2(SP)	:	LOW.CNT 6677
13969	047654	005003			CLR	R3	:	HIGH.CNT 6678
13970	047656	005066	000004		CLR	4(SP)	:	GTR.FLG 6679
13971	047662	005005			CLR	R5	:	BIT.CNT 6681
13972	047664	010504		3S:	MOV	R5, R4	:	BIT.CNT, * 6683
13973	047666	006204			ASR	R4	:	
13974	047670	006204			ASR	R4	:	
13975	047672	006204			ASR	R4	:	
13976	047674	012702	000010		MOV	#10, R2	:	
13977	047700	060602			ADD	SP, R2	:	PROM.DATA, * 6681
13978	047702	060204			ADD	R2, R4	:	
13979	047704	010446			MOV	R4, -(SP)	:	

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Label	Instruction	Comments	Line No.
13981						:ML4			
13982						:			
13983									
13984	047706	010546					MOV R5, -(SP)	: BIT.CNT,*	
13985	047710	042716	177770				BIC #177770, (SP)		
13986	047714	012746	000001				MOV #1, -(SP)		
13987	047720	005046					CLR -(SP)		
13988	047722	004767	133212				JSR PC, BL\$GT2		
13989	047726	062706	000010				ADD #10, SP		
13990	047732	005300					DEC R0		
13991	047734	001002					BNE 4\$		
13992	047736	005266	000002				INC 2(SP)	: LOW.CNT	
13993	047742	005205				4\$:	INC R5	: BIT.CNT	6681
13994	047744	020527	000011				CMP R5, #11	: BIT.CNT,*	
13995	047750	003745					BLE 3\$		
13996	047752	016603	000010				MOV 10(SP), R3	: PROM.DATA,HIGH.CNT	6685
13997	047756	004203					ASR R3	: HIGH.CNT	
13998	047760	004203					ASR R3	: HIGH.CNT	
13999	047762	004303					SWAB R3	: HIGH.CNT	
14000	047764	042703	177770				BIC #177770, R3	: *,HIGH.CNT	
14001	047770	132766	000040	000011			BITB #40, 11(SP)	: *,PROM.DATA+1	6687
14002	047776	001401					BEQ 5\$		
14003	050000	005203					INC R3	: HIGH.CNT	
14004	050002	132766	000100	000011	5\$:		BITB #100, 11(SP)	: *,PROM.DATA+1	6689
14005	050010	001401					BEQ 6\$		
14006	050012	005203					INC R3	: HIGH.CNT	
14007	050014	005005				6\$:	CLR R5		6691
14008	050016	105766	000011				TSTB 11(SP)	: PROM.DATA+1	
14009	050022	100002					BPL 7\$		
14010	050024	005205					INC R5		
14011	050026	005203					INC R3	: HIGH.CNT	
14012	050030	006005				7\$:	ROR R5		6693
14013	050032	103003					BCC 8\$		
14014	050034	012766	000001	000004			MOV #1, 4(SP)	: *,GTR.FLG	
14015	050042	004767	144114			8\$:	JSR PC, DAT.DM.XFER		6695
14016	050046	152777	000100	143220			BISB #100, @ML, REG+120		6696
14017	050054	016605	000010				MOV 10(SP), R5	: PROM.DATA,*	6697
14018	050060	010577	143320				MOV R5, @ML, REG+230		
14019	050064	012777	000061	143062			MOV #61, @ML, REG		6698
14020	050072	152777	000020	143174			BISB #20, @ML, REG+120		6699
14021	050100	032777	040000	143126			BIT #40000, @ML, REG+60		6701
14022	050106	001500					BEQ 12\$		
14023	050110	026627	000004	000001			CMP 4(SP), #1	: GTR.FLG,*	6705
14024	050116	001042					BNE 11\$		
14025	050120	010516					MOV R5, (SP)	: *,TEMP	6708
14026	050122	042716	015777				BIC #15777, (SP)	: *,TEMP	6709
14027	050126	021627	162000				CMP (SP), #-16000	: TEMP,*	6711
14028	050132	001003					BNE 9\$		
14029	050134	026603	000002				CMP 2(SP), R3	: LOW.CNT,HIGH.CNT	6715
14030	050140	002002					BGE 10\$		
14031	050142	000167	000422			9\$:	JMP 16\$		
14032	050146	104455				10\$:	TRAP 55		6718
14033	050150	000110					.WORD 110		
14034	050152	010570					.WORD ARR.DAT		
14035	050154	024052					.WORD DUMPER		

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

```

14037 ;ML4
14038 ;
14039 ;
14040 050156 012746 006552 MOV #WORD.24,-(SP) ;
14041 050162 012746 006654 MOV #WORD.33,-(SP) ;
14042 050166 012746 006336 MOV #WORD.6,-(SP) ;
14043 050172 012746 006646 MOV #WORD.32,-(SP) ;
14044 050176 012746 007472 MOV #PHR.5,-(SP) ;
14045 050202 012746 006660 MOV #WORD.34,-(SP) ;
14046 050206 012746 006100 MOV #SIX.FMT,-(SP) ;
14047 050212 012746 000007 MOV #7,-(SP) ;
14048 050216 010600 MOV SP,R0 ; SP,*
14049 050220 104414 TRAP 14 ;
14050 050222 000556 BR 15$ ;
14051 050224 026603 000002 11$: CMP 2(SP),R3 ; LOW.CNT,HIGH.CNT 6720
14052 050230 001157 BNE 16$ ; 6729
14053 050232 104455 TRAP 55 ; 6732
14054 050234 000111 .WORD 111 ;
14055 050236 010570 .WORD ARR.DAT ;
14056 050240 024052 .WORD DUMPER ;
14057 050242 012746 006552 MOV #WORD.24,-(SP) ; 6733
14058 050246 012746 006654 MOV #WORD.33,-(SP) ;
14059 050252 012746 006336 MOV #WORD.6,-(SP) ;
14060 050256 012746 006646 MOV #WORD.32,-(SP) ;
14061 050262 012746 007472 MOV #PHR.5,-(SP) ;
14062 050266 012746 006660 MOV #WORD.34,-(SP) ;
14063 050272 012746 006100 MOV #SIX.FMT,-(SP) ;
14064 050276 012746 000007 MOV #7,-(SP) ;
14065 050280 010600 MOV SP,R0 ; SP,*
14066 050284 104414 TRAP 14 ;
14067 050286 000524 BR 15$ ;
14068 050310 026627 000004 000001 12$: CMP 4(SP),#1 ; GTR.FLG,* 6734
14069 050316 001067 BNE 14$ ; 6743
14070 050320 010516 MOV R5,(SP) ; *,TEMP 6746
14071 050322 042716 015777 BIC #15777,(SP) ; *,TEMP 6747
14072 050326 021627 162000 CMP (SP),#-16000 ; TEMP,* 6749
14073 050332 001427 BEQ 13$ ;
14074 050334 104455 TRAP 55 ; 6752
14075 050336 000112 .WORD 112 ;
14076 050340 010570 .WORD ARR.DAT ;
14077 050342 024052 .WORD DUMPER ;
14078 050344 012746 006552 MOV #WORD.24,-(SP) ; 6753
14079 050350 012746 006654 MOV #WORD.33,-(SP) ;
14080 050354 012746 006330 MOV #WORD.5,-(SP) ;
14081 050360 012746 006646 MOV #WORD.32,-(SP) ;
14082 050364 012746 007366 MOV #PHR.1,-(SP) ;
14083 050370 012746 006660 MOV #WORD.34,-(SP) ;
14084 050374 012746 006100 MOV #SIX.FMT,-(SP) ;
14085 050400 012746 000007 MOV #7,-(SP) ;
14086 050404 010600 MOV SP,R0 ; SP,*
14087 050406 104414 TRAP 14 ;
14088 050410 000463 BR 15$ ;
14089 050412 026603 000002 13$: CMP 2(SP),R3 ; LOW.CNT,HIGH.CNT 6754
14090 050416 002064 BGE 16$ ; 6759
14091 050420 104455 TRAP 55 ; 6762

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

6719

6720

6729

6732

6733

6734

6743

6746

6747

6749

6752

6753

6754

6759

6762


```

14093      ;ML4
14094      :
14095      :
14096 050422 000113      .WORD 113
14097 050424 010570      .WORD ARR.DAT
14098 050426 024052      .WORD DUMPER
14099 050430 012746 006552  MOV #WRD.24,-(SP)      ;
14100 050434 012746 006654  MOV #WRD.33,-(SP)      ;
14101 050440 012746 006330  MOV #WRD.5,-(SP)       ;
14102 050444 012746 006646  MOV #WRD.32,-(SP)      ;
14103 050450 012746 007366  MOV #PHR.1,-(SP)       ;
14104 050454 012746 006660  MOV #WRD.34,-(SP)      ;
14105 050460 012746 006100  MOV #SIX.FMT,-(SP)     ;
14106 050464 012746 000007  MOV #7,-(SP)           ;
14107 050470 010600      MOV SP,R0              ; SP,*
14108 050472 104414      TRAP 14
14109 050474 000431      BR 158
14110 050476 026603 000002 14$: CMP 2(SP),R3        ; LOW.CNT,HIGH.CNT
14111 050502 001432      BEQ 168
14112 050504 104455      TRAP 55
14113 050506 000154      .WORD 154
14114 050510 010570      .WORD ARR.DAT
14115 050512 024052      .WORD DUMPER
14116 050514 012746 006552  MOV #WRD.24,-(SP)      ;
14117 050520 012746 006654  MOV #WRD.33,-(SP)      ;
14118 050524 012746 006330  MOV #WRD.5,-(SP)       ;
14119 050530 012746 006646  MOV #WRD.32,-(SP)      ;
14120 050534 012746 007366  MOV #PHR.1,-(SP)       ;
14121 050540 012746 006660  MOV #WRD.34,-(SP)      ;
14122 050544 012746 006100  MOV #SIX.FMT,-(SP)     ;
14123 050550 012746 000007  MOV #7,-(SP)           ;
14124 050554 010600      MOV SP,R0              ; SP,*
14125 050556 104414      TRAP 14
14126 050560 012701 000001 15$: MOV #1,R1          ; *,ERR.FLG
14127 050564 062706 000020      ADD #20,SP
14128 050570 020127 000001 16$: CMP R1,#1          ; ERR.FLG,*
14129 050574 001014      BNE 178
14130 050576 010546      MOV R5,-(SP)           ;
14131 050600 012746 004562  MOV #FMT.7,-(SP)       ;
14132 050604 012746 000002  MOV #2,-(SP)           ;
14133 050610 010600      MOV SP,R0              ; SP,*
14134 050612 104414      TRAP 14
14135 050614 012766 000001 000014  MOV #1,14(SP)          ; *,DODU.FLG
14136 050622 062706 000006      ADD #6,SP
14137 050626 104467      TRAP 67
14138 050630 006000      ROR R0
14139 050632 103002      BHIS 188
14140 050634 000167 176754      JMP 28
14141 050640 005205 18$: INC R5              ;
14142 050642 001402      BEQ 198
14143 050644 000167 176740      JMP 18
14144 050650 026627 000006 000001 19$: CMP 6(SP),#1          ; DODU.FLG,*
14145 050656 001004      BNE 208
14146 050660 016700 142714      MOV ML,LUN,R0
14147 050664 104451      TRAP 51

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
14149 :ML4
14150 :
14151 :
14152 050666 104444
14153 050670 062706 000012 20$: TRAP 44
14154 050674 000207 : ADD #12,SP ;
: RTS PC
14155 :
14156 : Routine Size: 292 words
14157 : Maximum stack depth per invocation: 19 words
14162 :
14163 :
14167 :
14171 050676 T26::
14172 050676 004767 176664 1$: JSR PC,ST26 ;
14173 050702 104466 : TRAP 66
14174 050704 006000 : ROR R0
14175 050706 103773 : BLO 1$
14176 050710 000207 : RTS PC
14177 :
14178 : Routine Size: 6 words
14179 : Maximum stack depth per invocation: 0 words
14184 :
14185 :
14186 ; 6804 !<BLF/PAGE>
```

6626

6801

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (77)

14188 :ML4

14189 :

14190 :

14191 : 6805

14192 : 6806

14193 : 6807

14194 : 6808

14195 : 6809

14196 : 6810

14197 : 6811

14198 : 6812

14199 : 6813

14200 : 6814

14201 : 6815

14202 : 6816

14203 : 6817

14204 : 6818

14205 : 6819

14206 : 6820

14207 : 6821

14208 : 6822

14209 : 6823

14210 : 6824

14211 : 6825

14212 : 6826

14213 : 6827

14214 : 6828

14215 : 6829

14216 : 6830

14217 : 6831

14218 : 6832

14219 : 6833

14220 : 6834

14221 : 6835

14222 : 6836

14223 : 6837

14224 : 6838

14225 : 6839

14226 : 6840

14227 : 6841

14228 : 6842

14229 : 6843

14230 : 6844

14231 : 6845

14232 : 6846

14233 : 6847

14234 : 6848

14235 : 6849

14236 : 6850

14237 : 6851

14238 : 6852

14239 : 6853

14240 : 6854

14241 : 6855

14242 : 6856

6805

6806

6807

6808

6809

6810

6811

6812

6813

6814

6815

6816

6817

6818

6819

6820

6821

6822

6823

6824

6825

6826

6827

6828

6829

6830

6831

6832

6833

6834

6835

6836

6837

6838

6839

6840

6841

6842

6843

6844

6845

6846

6847

6848

6849

6850

6851

6852

6853

6854

6855

6856

BGNTST;

!++

TEST NUMBER: TST 27

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.

Local

TST_PAT,
BAD_NIB_CNT,
ERR_FLG;

!TEST PATTERN
!NUMBER OF BAD NIBBLES FOUND
!ERROR FLAG

TST_PAT = ONES;
BAD_NIB_CNT = ZEROES;

incr TWICE from 0 to 1 do

!REPEAT LOOP TWICE

begin
BGNSUB;
CLR MBUS;
MLD1 = .TST_PAT;
MLD2 = .TST_PAT;
MLE2 = .TST_PAT;

!LOAD TEST PATTERN INTO DIAG REGISTERS

```

14244 :ML4
14245 :
14246 :
14247 :      6857      DAT_DM = ONE;
14248 :      6858      FIRST_BLK_XFER ();
14249 :      6859      MLCS1 = write;
14250 :      6860
14251 :      6861      incr CNT from 0 to 4 do
14252 :      6862      begin
14253 :      6863      DELAY (ONE_US);
14254 :      6864      DAT_CLK = ONE;
14255 :      6865      end;
14256 :      6866
14257 :      6867      CLR_MBUS;
14258 :      6868      DAT_DM = ONE;
14259 :      6869      FIRST_BLK_XFER ();
14260 :      6870      MLCS1 = read;
14261 :      6871      DELAY (ONE_US);
14262 :      6872
14263 :      6873      incr ARR_WRD from 0 to 4 do
14264 :      6874      begin
14265 :      6875      DAT_CLK = ONE;
14266 :      6876      DELAY (ONE_US);
14267 :      6877      RD_LNG_WRD;
14268 :      6878
14269 :      6879      incr NIB_PTR from 0 to 9 do
14270 :      6880      begin
14271 :      6881      TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG); !COMPARE TST PAT TO NIBBLE UNDER TEST
14272 :      6882
14273 :      6883      if .ERR_FLG !S_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;
14274 :      6884
14275 :      6885      !INCREMENT BAD_NIBBLE COUNT IFERR_FLG SET
14276 :      6886
14277 :      6887      end;
14278 :      6888      end;
14279 :      6889
14280 :      6890      TST_PAT = not .TST_PAT;
14281 :      6891      ENDSUB;
14282 :      6892      end;
14283 :      6893
14284 :      6894      if .BAD_NIB_CNT gtr 36
14285 :      6895      then
14286 :      6896      begin
14287 :      6897      ERRDF (77, ASYNC, DUMPER);
14288 :      6898      PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, WRD_45, FNC_14);
14289 :      6899      DODU (.ML_LDN);
14290 :      6900      DOCLN;
14291 :      6901      end;
14292 :      6902
14293 :      6903      ENDTST;
14297 :

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (77)

```

!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 XFERRERD WHERE BAD

!ERROR IF GTR 36

```

Address	OpCode	Op1	Op2	Op3	Label	Instruction	Comments	Line No.
14299					:ML4			
14300					:			
14301					:			
14305	050712	004167	133200		ST27:	JSR R1,SSAVE5	:	6803
14306	050716	024646				CMP -(SP),-(SP)	:	
14307	050720	012701	177777			MOV #1,R1	: *,TST.PAT	6847
14308	050724	005046				CLR -(SP)	: BAD.NIB.CNT	6848
14309	050728	005005				CLR R5	: TWICE	6850
14310	050730	104402			1S:	TRAP 2	:	6851
14311	050732	152777	000040	142254		BISB #40,@ML.REG+40	:	6852
14312	050740	016704	142636			MOV ML,DUT,R4		
14313	050744	042704	177770			BIC #177770,R4		
14314	050750	142777	000007	142236		BICB #7,@ML.REG+40		
14315	050756	150477	142232			BISB R4,@ML.REG+40		
14316	050762	010177	142356			MOV R1,@ML.REG+170	: TST.PAT,*	6854
14317	050766	010177	142362			MOV R1,@ML.REG+200	: TST.PAT,*	6855
14318	050772	010177	142336			MOV R1,@ML.REG+160	: TST.PAT,*	6856
14319	050776	152777	000010	142270		BISB #10,@ML.REG+120		6857
14320	051004	004767	145030			JSR PC,FIRST.BLK.XFER		6858
14321	051010	012777	000061	142136		MOV #61,@ML.REG		6859
14322	051016	005002				CLR R2	: CNT	6861
14323	051020	012703	000001		2S:	MOV #1,R3	: *,SSTMP2	6863
14324	051024	001411			3S:	BEQ 6S		
14325	051026	016704	131064			MOV LSDLY,R4	: *,SSTMP1	
14326	051032	001404				BEQ 5S		
14327	051034	005066	000004		4S:	CLR 4(SP)	: SSTMP	
14328	051040	005304				DEC R4	: SSTMP1	
14329	051042	001374				BNE 4S		
14330	051044	005303			5S:	DEC R3	: SSTMP2	
14331	051046	000766				BR 3S		
14332	051050	152777	000020	142216	6S:	BISB #20,@ML.REG+120	:	6864
14333	051056	005202				INC R2	: CNT	6861
14334	051060	020227	000004			CMP R2,#4	: CNT,*	
14335	051064	003755				BLE 2S		
14336	051066	152777	000040	142120		BISB #40,@ML.REG+40	:	6865
14337	051074	016704	142502			MOV ML,DUT,R4		
14338	051100	042704	177770			BIC #177770,R4		
14339	051104	142777	000007	142102		BICB #7,@ML.REG+40		
14340	051112	150477	142076			BISB R4,@ML.REG+40		
14341	051116	152777	000010	142150		BISB #10,@ML.REG+120		6868
14342	051124	04767	144710			JSR PC,FIRST.BLK.XFER		6869
14343	051130	012777	000071	142016		MOV #71,@ML.REG		6870
14344	051136	012703	000001			MOV #1,R3	: *,SSTMP2	6871
14345	051142	001411			7S:	BEQ 10S		
14346	051144	016704	130746			MOV LSDLY,R4	: *,SSTMP1	
14347	051150	001404				BEQ 9S		
14348	051152	005066	000004		8S:	CLR 4(SP)	: SSTMP	
14349	051156	005304				DEC R4	: SSTMP1	
14350	051160	001374				BNE 8S		
14351	051162	005303			9S:	DEC R3	: SSTMP2	
14352	051164	000766				BR 7S		
14353	051166	005002			10S:	CLR R2	: ARR.WRD	6873

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4


```
14411 ;ML4
14412 ;
14413 ;
14414 051432 016700 142142      MOV    ML.LUN,R0      ;
14415 051436 104451             TRAP   51             ;
14416 051440 104444             TRAP   44             ;
14417 051442 062706 000016      ADD    #16,SP         ;
14418 051446 062706 000006      20$:  ADD    #6,SP         ;
14419 051452 000207             RTS     PC             ;
14420 ;
14421 ; Routine Size: 177 words
14422 ; Maximum stack depth per invocation: 16 words
14423 ;
14424 ;
14425 ;
14426 ;
14427 ;
14428 ;
14429 ;
14430 ;
14431 ;
14432 ;
14433 ;
14434 ;
14435 ;
14436 051454             T27::
14437 051454 004767 177232      1$:   JSR    PC,$T27      ;
14438 051460 104466             TRAP   66             ;
14439 051462 006000             ROR    R0             ;
14440 051464 103773             BLO    1$             ;
14441 051466 000207             RTS     PC             ;
14442 ;
14443 ; Routine Size: 6 words
14444 ; Maximum stack depth per invocation: 0 words
14445 ;
14446 ;
14447 ;
14448 ;
14449 ;
14450 ;
14451 :           6904  !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

6899
6896
6803

6901

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (78)

```

14453 :ML4
14454 :
14455 :
14456 : 6905 !
14457 : 6906 ! BGNTST;
14458 : 6907 !
14459 : 6908 ! ++
14460 : 6909 ! TEST NUMBER: TST 28
14461 : 6910 !
14462 : 6911 ! TEST NAME: PROM SELECTION TEST
14463 : 6912 !
14464 : 6913 ! TEST DESCRIPTION:
14465 : 6914 ! DUE TO THE NATURE OF THE DEVICE
14466 : 6915 ! AND OF THE ARRAY MODULES' UV
14467 : 6916 ! PROMS, ONLY PROM READS ARE
14468 : 6917 ! ALLOWED DURING DIAG TESTING.
14469 : 6918 !
14470 : 6919 ! THEREFORE THE ARRAY MODULE UV PROMS
14471 : 6920 ! ARE TESTED FOR UNIQUE SELECTION BY:
14472 : 6921 !
14473 : 6922 ! 1. AT EACH PRESENT ARRAY MODULE WRITE 127 ARRAY WORDS WITH 1'S/0'S PATTERN.
14474 : 6923 !
14475 : 6924 ! 2. READ THE UV PROMS AT THEIR RESPECTIVE ARRAY WORD LOCATION AND SEE IF
14476 : 6925 ! THE PROMS MASK BAD NIBBLE LOCATIONS (ENCOUNTERED BAD NIBBLES
14477 : 6926 ! INDICATES INCORRECT MASKING). COUNT EACH BAD NIBBLE ENCOUNTERED AT AN
14478 : 6927 ! ARRAY MODULE.
14479 : 6928 !
14480 : 6929 ! 3. ALLOW A THRESHOLD OF 5 BAD NIBBLES AT ANY ARRAY MODULE.
14481 : 6930 !
14482 : 6931 ! 4. REPORT PROM SEL ERRORS AT RESPECTIVE ARRAY MODULE IF THE
14483 : 6932 ! THRESHOLD IS EXCEEDED.
14484 : 6933 !
14485 : 6934 ! IMPLICIT INPUTS:
14486 : 6935 ! PD TEMP:
14487 : 6936 ! A BIT VECTOR OF 16 BITS WHERE
14488 : 6937 ! THE READ PROM DATA IS STORED
14489 : 6938 ! AND ACCESSED FROM.
14490 : 6939 !
14491 : 6940 ! IO BUF
14492 : 6941 ! A VECTOR OF 256 WORDS WHERE
14493 : 6942 ! DATA FOR MBUS READS AND WRITE
14494 : 6943 ! FUNCTION ARE FOUND.
14495 : 6944 !
14496 : 6945 !
14497 : 6946 !
14498 : 6947 ! local
14499 : 6948 ! DODU_FLG, !DROP UNIT FLG
14500 : 6949 ! ERR_FLG, !ERROR FLG
14501 : 6950 ! TST_PAT, !TEST PATTERN
14502 : 6951 ! ERR_CNT; !ERROR COUNT
14503 : 6952 !
14504 : 6953 ! DODU_FLG = ZERO;
14505 : 6954 ! TST_PAT = ONES;
14506 : 6955 !
14507 : 6956 ! incr ARR_SEL from 0 to .LST_ARR by .ARR_INC do !TEST ALL PRESENT ARRAYS

```


22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (78)

```

14509 :ML4
14510 :
14511 :
14512 :      6957      begin
14513 :      6958      ERR CNT = ZERO;
14514 :      6959      BGNSUB;
14515 :      6960
14516 :      6961      incr TWICE from 0 to 1 do      !REPEAT LOOP TWICE
14517 :      6962      begin
14518 :      6963      CLR_MBUS;
14519 :      6964      MLD1 = .TST_PAT;      !LOAD DATA DIAG REGISTERS WITH TST_PAT
14520 :      6965      MLD2 = .TST_PAT;
14521 :      6966      MLD3 = .TST_PAT;
14522 :      6967      DAT_DM = ONE;      !SET DATA DIAG MODE
14523 :      6968      MLWC = not 255;      !LOAD WORD COUNT
14524 :      6969      MLBA = IO BUF;      !LOAD UBUS ADRS
14525 :      6970      MLDA = .ARR_SEL;      !LOAD SECTOR
14526 :      6971      MLCS1 = write;      !DO A WRITE FUNCTION
14527 :      6972
14528 :      6973      incr CNT from 0 to 127 do      !CLOCK IN 127 WORDS
14529 :      6974      begin
14530 :      6975      DELAY (ONE_US);
14531 :      6976      DAT_CLK = ONE;
14532 :      6977      end;
14533 :      6978
14534 :      6979      CLR_MBUS;
14535 :      6980      DAT_DM = ONE;      !SET DATA DIAG MODE
14536 :      6981      MLWC = not 255;      !LOAD WORD COUNT
14537 :      6982      MLBA = IO BUF;      !LOAD UBUS ADRS
14538 :      6983      MLDA = .ARR_SEL;      !LOAD SECTOR
14539 :      6984      MLCS1 = read;      !DO A READ FUNCTION
14540 :      6985      DELAY (ONE_US);
14541 :      6986
14542 :      6987      incr WD_CNT from 0 to 127 do      !READ THE 10 WORDS
14543 :      6988      begin
14544 :      6989      PD_TEMP = .MLPD;      !GET PROM DATA FOR THIS WORD
14545 :      6990      DAT_CLK = ONE;      !CLOCK THIS WORD INTO DIAG REG
14546 :      6991      DELAY (ONE_US);
14547 :      6992      RD_LNG_WRD;      !READ DIAG REG FOR THIS WORD
14548 :      6993
14549 :      6994      incr NIB_PTR from 0 to 9 do      !LOOK AT ALL 10 NIBBLE
14550 :      6995
14551 :      6996      if .PD_TEMP [.NIB_PTR] IS_NOT_SET      !FIND GOOD NIBBLES
14552 :      6997      then
14553 :      6998      begin
14554 :      6999      TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG); !COMPARE NIBBLE TO TST_PAT
14555 :      7000
14556 :      7001      if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1;
14557 :      7002
14558 :      7003      !INCREMENT ERROR COUNT IF ERROR FLG IS SET
14559 :      7004      end;
14560 :      7005
14561 :      7006      end;
14562 :      7007
14563 :      7008      TST_PAT = not .TST_PAT;      !REPEAT WITH COMPLIMENT DATA

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (78)

```

14565 :ML4
14566 :
14567 :
14568 :       7009           end;
14569 :       7010
14570 :       7011           if .ERR_CNT gtr 5           !ALLOW 5 ERROR BEFORE ERRORING
14571 :       7012           then
14572 :       7013           begin                       !ERROR IF GTR 5
14573 :       7014           ERRDF (78, ARR DAT, DUMPER);
14574 :       7015           PRINTB (THR_FMT, WRD_35, WRD_37, WRD_10);
14575 :       7016           PRINTB (FMT_9, .ARR_SEL);
14576 :       7017           DODU_FLG = ONE;
14577 :       7018           end;
14578 :       7019
14579 :       7020           ENDSUB;
14580 :       7021
14581 :       7022           if .DODU_FLG IS_SET           !DROP THIS UNIT IF DODU_FLG IS SET
14582 :       7023           then
14583 :       7024           begin
14584 :       7025           DODU (.ML_LUN);
14585 :       7026           DOCLN;
14586 :       7027           end;
14587 :       7028
14588 :       7029           end;
14589 :       7030
14590 :       7031           ENDTST;
14594 :

```

14598	051470	004167	132422	\$T28:	JSR	R1,\$\$SAVE5	:	6903	
14599	051474	162706	000016		SUB	#16,SP	:		
14600	051500	005066	000010		CLR	10(SP)	:	DODU.FLG	
14601	051504	012702	177777		MOV	#-1,R2	:	*,TST.PAT	
14602	051510	016766	140062	000006	MOV	LST.ARR,6(SP)	:	6954	
14603	051516	016766	140040	000004	MOV	ARR.INC,4(SP)	:	6956	
14604	051524	005001			CLR	R1	:	ARR.SEL	
14605	051526	000167	000666		JMP	25\$:		
14606	051532	005066	000002	1\$:	CLR	2(SP)	:	ERR.CNT	
14607	051536	104402		2\$:	TRAP	2	:	6958	
14608	051540	005016			CLR	(SP)	:	TWICE	
14609	051542	152777	000040	141444	3\$:	BISB	#40,@ML.REG+40	:	6961
14610	051550	016705	142026		MOV	ML,DUT,R5	:	6962	
14611	051554	042705	177770		BIC	#177770,R5	:		
14612	051560	142777	000007	141426	BICB	#7,@ML.REG+40	:		
14613	051566	150577	141422		BISB	R5,@ML.REG+40	:		
14614	051572	010277	141546		MOV	R2,@ML.REG+170	:	TST.PAT,*	
14615	051576	010277	141552		MOV	R2,@ML.REG+200	:	TST.PAT,*	
14616	051602	010277	141526		MOV	R2,@ML.REG+160	:	TST.PAT,*	
14617	051606	152777	000010	141460	BISB	#10,@ML.REG+120	:	6964	
14618	051614	012777	177400	141342	MOV	#-400,@ML.REG+10	:	6965	
							:	6966	
							:	6967	
							:	6968	

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

```

14620                :ML4
14621                :
14622                :
14623 051622 012777 011600 141344        MOV      #10,BUF,@ML.REG+20
14624 051630 010177 141350                MOV      R1,@ML.REG+30
14625 051634 012777 000061 141312        MOV      #61,@ML.REG
14626 051642 005003                    CLR      R3
14627 051644 012704 000001            4S:    MOV      #1,R4
14628 051650 001411                    5S:    BEQ      8S
14629 051652 016705 130240                MOV      LSDLY,R5
14630 051656 001404                    BEQ      7S
14631 051660 005066 000014            6S:    CLR      14(SP)
14632 051664 005305                    DEC      R5
14633 051666 001374                    BNE     6S
14634 051670 005304            7S:    DEC      R4
14635 051672 000766                    BR       5S
14636 051674 152777 000020 141372    8S:    BISB   #20,@ML.REG+120
14637 051702 005203                    INC      R3
14638 051704 020327 000177                CMP      R3,#177
14639 051710 003755                    BLE     4S
14640 051712 152777 000040 141274    BISB   #40,@ML.REG+40
14641 051720 016705 141656                MOV      ML,DUT,R5
14642 051724 042705 177770                BIC     #177770,R5
14643 051730 142777 000007 141256    BICB   #7,@ML.REG+40
14644 051736 150577 141252                BISB   R5,@ML.REG+40
14645 051742 152777 000010 141324    BISB   #10,@ML.REG+120
14646 051750 012777 177400 141206    MOV     #400,@ML.REG+10
14647 051756 012777 011600 141210    MOV     #10,BUF,@ML.REG+20
14648 051764 010177 141214                MOV      R1,@ML.REG+30
14649 051770 012777 000071 141156    MOV     #71,@ML.REG
14650 051776 012704 000001                MOV     #1,R4
14651 052002 001411            9S:    BEQ     12S
14652 052004 016705 130106                MOV     LSDLY,R5
14653 052010 001404                    BEQ     11S
14654 052012 005066 000014            10S:   CLR     14(SP)
14655 052016 005305                    DEC     R5
14656 052020 001374                    BNE    10S
14657 052022 005304            11S:   DEC     R4
14658 052024 000766                    BR     9S
14659 052026 005003            12S:   CLR     R3
14660 052030 017767 141350 141062    13S:   MOV     @ML.REG+230,PD.TEMP
14661 052036 152777 000020 141230    BISB   #20,@ML.REG+120
14662 052044 012704 000001                MOV     #1,R4
14663 052050 001411            14S:   BEQ     17S
14664 052052 016705 130040                MOV     LSDLY,R5
14665 052056 001404                    BEQ     16S
14666 052060 005066 000014            15S:   CLR     14(SP)
14667 052064 005305                    DEC     R5
14668 052066 001374                    BNE    15S
14669 052070 005304            16S:   DEC     R4
14670 052072 000766                    BR     14S
14671 052074 017767 141244 137050    17S:   MOV     @ML.REG+170,D1.TEMP
14672 052102 017767 141246 137044    MOV     @ML.REG+200,D2.TEMP
14673 052110 017767 141220 137040    MOV     @ML.REG+160,E2.TEMP
14674 052116 005004                    CLR     R4

```

```

: ARR.SEL,*
: CNT
*:SSTMP2
*:SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: CNT
: CNT,*
:
:
:
:
: ARR.SEL,*
*:SSTMP2
*:SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: WD.CNT
:
*:SSTMP2
*:SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: NIB.PTR

```

```

6969
6970
6971
6973
6975
6976
6975
6977
6980
6981
6982
6983
6984
6985
6987
6989
6990
6991
6994

```

14676															
14677					:ML4										
14678					:										
14679	052120	010405			18\$:	MOV	R4,R5			: NIB.PTR,*					6996
14680	052122	006205				ASR	R5								
14681	052124	006205				ASR	R5								
14682	052126	006205				ASR	R5								
14683	052130	062705	013120			ADD	#PD.TEMP,R5								
14684	052134	010546				MOV	R5,-(SP)								
14685	052136	010446				MOV	R4,-(SP)			: NIB.PTR,*					
14686	052140	042716	177770			BIC	#177770,(SP)								
14687	052144	012746	000001			MOV	#1,-(SP)								
14688	052150	005046				CLR	-(SP)								
14689	052152	004767	130762			JSR	PC,BLSGT2								
14690	052156	062706	000010			ADD	#10,SP								
14691	052162	005700				TST	R0								
14692	052164	001017				BNE	20\$								
14693	052166	010446				MOV	R4,-(SP)			: NIB.PTR,*					6999
14694	052170	010246				MOV	R2,-(SP)			: TST.PAT,*					
14695	052172	012746	000020			MOV	#20,-(SP)								
14696	052176	060616				ADD	SP,(SP)			: ERR.FLG,*					
14697	052200	004767	143742			JSR	PC,TST.LNG.WRD								
14698	052204	026627	000020	000001		CMP	20(SP),#1			: ERR.FLG,*					7001
14699	052212	001002				BNE	19\$								
14700	052214	005266	000010			INC	10(SP)			: ERR.CNT					
14701	052220	062706	000006			ADD	#6,SP								6998
14702	052224	005204			19\$:	INC	R4			: NIB.PTR					6994
14703	052226	020427	000011		20\$:	CMP	R4,#11			: NIB.PTR,*					
14704	052232	003732				BLE	18\$								
14705	052234	005203				INC	R3			: WD.CNT					6987
14706	052236	020327	000177			CMP	R3,#177			: WD.CNT,*					
14707	052242	003672				BLE	13\$								
14708	052244	005102				COM	R2			: TST.PAT					7008
14709	052246	005216				INC	(SP)			: TWICE					6961
14710	052250	021627	000001			CMP	(SP),#1			: TWICE,*					
14711	052254	003002				BGT	21\$								
14712	052256	000167	177260			JMP	3\$								
14713	052262	026627	000002	000005	21\$:	CMP	2(SP),#5			: ERR.CNT,*					7011
14714	052270	003434				BLE	22\$								
14715	052272	104455				TRAP	5\$:					7014
14716	052274	000116				.WORD	116								
14717	052276	010570				.WORD	ARR.DAT								
14718	052300	024052				.WORD	DUMPER								
14719	052302	012746	006406			MOV	#WRD.10,-(SP)			:					7015
14720	052306	012746	006700			MOV	#WRD.37,-(SP)								
14721	052312	012746	006666			MOV	#WRD.35,-(SP)								
14722	052316	012746	006034			MOV	#THR.FMT,-(SP)								
14723	052322	012746	000004			MOV	#4,-(SP)								
14724	052326	010600				MOV	SP,R0			: SP,*					
14725	052330	104414				TRAP	14								
14726	052332	010116				MOV	R1,(SP)			: ARR.SEL,*					7016
14727	052334	012746	004644			MOV	#FMT.9,-(SP)								
14728	052340	012746	000002			MOV	#2,-(SP)								
14729	052344	010600				MOV	SP,R0			: SP,*					
14730	052346	104414				TRAP	14								

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

14732      ;ML4
14733      ;
14734      ;
14735 052350 012766 000001 000026      MOV      #1,26(SP)      ; *,DODU.FLG      7017
14736 052356 062706 000016      ADD      #16,SP      ;      7018
14737 052362 104467      22$: TRAP      67      ;      7018
14738 052364 006000      ROR      R0      ;
14739 052366 103002      BHIS     23$      ;
14740 052370 000167 177142      JMP      2$      ;
14741 052374 026627 000010 000001 23$: CMP      10(SP),#1      ; DODU.FLG,*      7022
14742 052402 001004      BNE     24$      ;
14743 052404 016700 141170      MOV      ML,LUN,R0      ;      7025
14744 052410 104451      TRAP     51      ;
14745 052412 104444      TRAP     44      ;
14746 052414 066601 000004 24$: ADD      4(SP),R1      ; *,ARR.SEL      6956
14747 052420 020166 000006 25$: CMP      R1,6(SP)      ; ARR.SEL,*
14748 052424 003002      BGT     26$      ;
14749 052426 000167 177100      JMP      1$      ;
14750 052432 062706 000016 26$: ADD      #16,SP      ;      6903
14751 052436 000207      RTS     PC      ;
14752
14753      ; Routine Size: 244 words
14754      ; Maximum stack depth per invocation: 20 words
14759
14760
14764
14768 052440      T28::
14769 052440 004767 177024 1$: JSR      PC,$T28      ;      7029
14770 052444 104466      TRAP     66      ;
14771 052446 006000      ROR      R0      ;
14772 052450 103773      BLO     1$      ;
14773 052452 000207      RTS     PC      ;
14774
14775      ; Routine Size: 6 words
14776      ; Maximum stack depth per invocation: 0 words
14781
14782
14783 :      7032 !<BLF/PAGE>

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (79)

14785 :ML4
14786 :
14787 :
14788 :
14789 :
14790 :
14791 :
14792 :
14793 :
14794 :
14795 :
14796 :
14797 :
14798 :
14799 :
14800 :
14801 :
14802 :
14803 :
14804 :
14805 :
14806 :
14807 :
14808 :
14809 :
14810 :
14811 :
14812 :
14813 :
14814 :
14815 :
14816 :
14817 :
14818 :
14819 :
14820 :
14821 :
14822 :
14823 :
14824 :
14825 :
14826 :
14827 :
14828 :
14829 :
14830 :
14831 :
14832 :
14833 :
14834 :
14835 :
14836 :
14837 :
14838 :
14839 :

7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068
7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084

```
!
!BGNTST;
!++
!TEST NUMBER: TST 29
!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
```

14841 :ML4
14842 :
14843 :
14844 :
14845 :
14846 :
14847 :
14848 :
14849 :
14850 :
14851 :
14852 :
14853 :
14854 :
14855 :
14856 :
14857 :
14858 :
14859 :
14860 :
14861 :
14862 :
14863 :
14864 :
14865 :
14866 :
14867 :
14868 :
14869 :
14870 :
14871 :
14872 :
14873 :
14874 :
14875 :
14876 :
14877 :
14878 :
14879 :
14880 :
14881 :
14882 :
14883 :
14884 :
14885 :
14886 :
14887 :
14888 :
14889 :
14890 :
14891 :
14892 :
14893 :
14894 :
14895 :

```

7085     PASS CNT,
7086     NIB PAT,
7087     DONE FLG,
7088     ERR FLG,
7089     SECTOR_NO;
7090
7091     BGNSUB;
7092     PASS CNT = -1;
7093     SECTOR NO = ZEROES;
7094     DONE_FLG = ZERO;
7095
7096     do
7097     begin
7098
7099     do
7100     begin
7101     PASS_CNT = .PASS_CNT + 1;
7102     CLR_MBUS;
7103     DAT_DM = ONE;
7104     MLDA = .SECTOR_NO;
7105     MLWC = not 255;
7106     MLBA = IO_BUF;
7107
7108     case .PASS_CNT from 0 to 3 of
7109     set
7110
7111     [0] :
7112         NIB_PAT = %0'000000';
7113
7114     [1] :
7115         NIB_PAT = %0'17';
7116
7117     [2] :
7118         NIB_PAT = %0'12';
7119
7120     [3] :
7121         NIB_PAT = %0'15'
7122     tes;
7123
7124     incr LD_CNT from 0 to 9 do
7125     LD_LNG_WRD (.LD_CNT, .NIB_PAT);
7126
7127     WRT_LNG_WRD;
7128     MLC51 = write;
7129
7130     incr WRT_CNT from 0 to 127 do
7131     begin
7132     DELAY (ONE_US);
7133     DAT_CLK = ONE;
7134     end;
7135
7136     CLR_MBUS;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (79)

```

!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 MBUS 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

```

14897 :ML4
14898 :
14899 :
14900 :
14901 :
14902 :
14903 :
14904 :
14905 :
14906 :
14907 :
14908 :
14909 :
14910 :
14911 :
14912 :
14913 :
14914 :
14915 :
14916 :
14917 :
14918 :
14919 :
14920 :
14921 :
14922 :
14923 :
14924 :
14925 :
14926 :
14927 :
14928 :
14929 :
14930 :
14931 :
14932 :
14933 :
14934 :
14935 :
14936 :
14937 :
14938 :
14939 :
14940 :
14941 :
14942 :
14943 :
14944 :
14945 :
14946 :
14947 :
14948 :
14949 :
14950 :
14951 :

7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188

```

DAT_DM = ONE;
MLDA = .SECTOR_NO;
MLWC = not 255;
MLBA = IO_BUF;
MLCS1 = read;
DELAY (ONE_US);
WRD_CNT = -1;

do
begin
WRD_CNT = .WRD_CNT + 1;
PD_TEMP = .MLPD;
DAT_CLK = ONE;
DELAY (ONE_US);
RD_LNG_WRD;
NIB_PTR = -1;

do
begin
NIB_PTR = .NIB_PTR + 1;

if .PD_TEMP [.NIB_PTR] IS_NOT_SET      !TEST THIS NIB IF PROM FLAG NOT SET
then
TST_LNG_WRD (.NIB_PTR, .NIB_PAT, ERR_FLG);

end
until (.ERR_FLG) or (.NIB_PTR eql 9);

end
until (.ERR_FLG) or (.WRD_CNT eql 127);

end
until (.PASS_CNT eql 3) or (.ERR_FLG IS_SET );      !REPEAT UNTIL ALL PAT TESTED OR ERROR FLG GETS SET
if (.PASS_CNT eql 3) and (.ERR_FLG IS_NOT_SET )    !WAS THIS A GOOD BLOCK?
then
begin
DONE_FLG = ONE;      !YES
GOOD_BLK = .SECTOR_NO;      !SET DONE FLAG
end
else
begin
SECTOR_NO = .SECTOR_NO + .RAS_INC;      !NO
PASS_CNT = -1;      !INCREMENT ROW NO
end;      !RESET PASS COUNT

end
until (.DONE_FLG IS_SET ) or (.SECTOR_NO eql .LST_ARR + .ARR_INC);

ENDSUB;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (79)

```

!SET DATA DIAG MODE
!LOAD SECTOR NUMBER
!LOAD WORD COUNT
!LOAD UBUS ADRS
!DO A READ FUNCTION

!RESET THE WORD COUNT

!TEST ONE BLOCK FOR NIBBLE PATTERN

!GET PROM DATA
!CLOCK DATA WORD INTO DIAG REG
!READ DIAG REGISTERS
!RESET THE NIBBLE POINTER

!SEARCH FOR GOOD NIBBLES
!INCREMENT NIBBLE POINTER

!REPEAT UNTIL ALL PAT TESTED OR ERROR FLG GETS SET
!WAS THIS A GOOD BLOCK?
!YES
!SET DONE FLAG
!GOOD BLOCK GETS THIS SECTOR NO
!NO
!INCREMENT ROW NO
!RESET PASS COUNT
!REPEAT UNTIL GOOD BLK FOUND OR AT LST ROW

```


22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (79)

```

14953 :ML4
14954 :
14955 :
14956 :       7189 if .SECTOR_NO eql .LST_ARR + .ARR_INC
14957 :       7190 then
14958 :       7191   begin
14959 :       7192     ERRDF (79, INTER, DUMPER);
14960 :       7193     PRINTB (THR_FMT, WRD_14, PHR_10, FNC_15);
14961 :       7194     DODU (.ML_LDN);
14962 :       7195     DOCLN;
14963 :       7196     end;
14964 :
14965 :       7197
14966 :
14967 :       7198 ENDTST;

```

!SEE IF WE'RE AT THE LAST BLOCK
!A GOOD BLK NOT FOUND BEFORE LAST BLK
!ERROR AND DROP UNIT

```

14973 052454 004167 131436      $T29: JSR    R1,$$SAVE5      :
14974 052460 162706 000014      SUB    #14,SP          :
14975 052464 104402          1$: TRAP  2              :
14976 052466 012766 177777 000002  MOV    #-1,2(SP)      : * ,PASS.CNT
14977 052474 005002          CLR    R2              : SECTOR.NO
14978 052476 005066 000004          CLR    4(SP)          : DONE.FLG
14979 052502 005266 000002          2$: INC    2(SP)          : PASS.CNT
14980 052506 152777 000040 140500  BISB  #40,@ML.REG+40
14981 052514 016705 141062          MOV    ML.DUT,R5
14982 052520 042705 177770          BIC   #177770,R5
14983 052524 142777 000007 140462  BICB  #7,@ML.REG+40
14984 052532 150577 140456          BISB  R5,@ML.REG+40
14985 052536 152777 000010 140530  BISB  #10,@ML.REG+120
14986 052544 010277 140434          MOV    R2,@ML.REG+30
14987 052550 012777 177400 140406  MOV    #-400,@ML.REG+10
14988 052556 012777 011600 140410  MOV    #10,BUF,@ML.REG+20
14989 052564 016605 000002          MOV    2(SP),R5
14990 052570 006305          ASL   R5
14991 052572 066507 052576          ADD   3$(R5),PC
14992 052576 000010          3$: .WORD 4$-3$
14993 052600 000014          .WORD 5$-3$
14994 052602 000022          .WORD 6$-3$
14995 052604 000030          .WORD 7$-3$
14996 052606 005001          4$: CLR    R1              : NIB.PAT
14997 052610 000410          BR    8$              :
14998 052612 012701 000017          5$: MOV    #17,R1         : * ,NIB.PAT
14999 052616 000405          BR    8$              :
15000 052620 012701 000012          6$: MOV    #12,R1         : * ,NIB.PAT
15001 052624 000402          BR    8$              :
15002 052626 012701 000015          7$: MOV    #15,R1         : * ,NIB.PAT
15003 052632 005005          8$: CLR    R5              : LD.CNT
15004 052634 010546          9$: MOV    R5,-(SP)      : LD.CNT,*
15005 052636 010146          MOV    R1,-(SP)      : NIB.PAT,*
15006 052640 004767 144260          JSR   PC,LD.LNG.WRD

```

7031

7089

7092

7093

7094

7101

7103

7104

7105

7106

7108

7112

7108

7115

7108

7118

7108

7121

7124

7125

```
15008 ;ML4
15009 ;
15010
15011 052644 022626 CMP (SP)+,(SP)+
15012 052646 005205 INC R5 ; LD.CNT 7124
15013 052650 020527 000011 CMP R5,#11 ; LD.CNT,*
15014 052654 003767 BLE 9$
15015 052656 016777 136270 140460 MOV D1.TEMP,@ML.REG+170 ;
15016 052664 016777 136264 140462 MOV D2.TEMP,@ML.REG+200 ;
15017 052672 016777 136260 140434 MOV E2.TEMP,@ML.REG+160 ;
15018 052700 012777 000061 140246 MOV #61,@ML.REG ;
15019 052706 005003 CLR R3 ; WRT.CNT 7128
15020 052710 012704 000001 10$: MOV #1,R4 ; *,SSTMP2 7130
15021 052714 001411 11$: BEQ 14$ ; *,SSTMP2 7132
15022 052716 016705 127174 MOV L$DLY,R5 ; *,SSTMP1
15023 052722 001404 BEQ 13$
15024 052724 005066 000012 12$: CLR 12(SP) ; SSTMP
15025 052730 005305 DEC R5 ; SSTMP1
15026 052732 001374 BNE 12$
15027 052734 005304 13$: DEC R4 ; SSTMP2
15028 052736 000766 BR 11$
15029 052740 152777 000020 140326 14$: BISB #20,@ML.REG+120 ;
15030 052746 005203 INC R3 ; WRT.CNT 7133
15031 052750 020327 000177 CMP R3,#177 ; WRT.CNT,* 7130
15032 052754 003755 BLE 10$
15033 052756 152777 000040 140230 BISB #40,@ML.REG+40 ; 7134
15034 052764 016705 140612 MOV ML,DUT,R5
15035 052770 042705 177770 BIC #177770,R5
15036 052774 142777 000007 140212 BICB #7,@ML.REG+40
15037 053002 150577 140206 BISB R5,@ML.REG+40
15038 053006 152777 000010 140260 BISB #10,@ML.REG+120 ;
15039 053014 010277 140164 MOV R2,@ML.REG+30 ; SECTOR.NO,* 7137
15040 053020 012777 177400 140136 MOV #-400,@ML.REG+10 ; 7138
15041 053026 012777 011600 140140 MOV #10.BUF,@ML.REG+20 ; 7139
15042 053034 012777 000071 140112 MOV #71,@ML.REG ; 7140
15043 053042 012704 000001 MOV #1,R4 ; *,SSTMP2 7141
15044 053046 001411 15$: BEQ 18$ ; *,SSTMP2 7142
15045 053050 016705 127042 MOV L$DLY,R5 ; *,SSTMP1
15046 053054 001404 BEQ 17$
15047 053056 005066 000012 16$: CLR 12(SP) ; SSTMP
15048 053062 005305 DEC R5 ; SSTMP1
15049 053064 001374 BNE 16$
15050 053066 005304 17$: DEC R4 ; SSTMP2
15051 053070 000766 BR 15$
15052 053072 012766 177777 000006 18$: MOV #-1,6(SP) ; *,WRD.CNT 7143
15053 053100 005266 000006 19$: INC 6(SP) ; WRD.CNT 7147
15054 053104 017767 140274 140006 MOV @ML.REG+230,PD.TEMP ; 7148
15055 053112 152777 000020 140154 BISB #20,@ML.REG+120 ; 7149
15056 053120 012704 000001 MOV #1,R4 ; *,SSTMP2 7150
15057 053124 001411 20$: BEQ 23$ ; *,SSTMP2
15058 053126 016705 126764 MOV L$DLY,R5 ; *,SSTMP1
15059 053132 001404 BEQ 22$
15060 053134 005066 000012 21$: CLR 12(SP) ; SSTMP
15061 053140 005305 DEC R5 ; SSTMP1
15062 053142 001374 BNE 21$
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

15064      ;ML4
15065      ;
15066
15067 053144 005304      22$: DEC      R4          : $STMP2
15068 053146 000766      BR      20$
15069 053150 017767 140170 135774 23$: MOV      @ML.REG+170,D1.TEMP
15070 053156 017767 140172 135770      MOV      @ML.REG+200,D2.TEMP
15071 053164 017767 140144 135764      MOV      @ML.REG+160,E2.TEMP
15072 053172 012716 177777      MOV      #-1,(SP)
15073 053176 005216      24$: INC      (SP)          : *,NIB.PTR
15074 053200 011605      MOV      (SP),R5          : NIB.PTR
15075 053202 006205      ASR      R5              : NIB.PTR,*
15076 053204 006205      ASR      R5
15077 053206 006205      ASR      R5
15078 053210 062705 013120      ADD      #PD.TEMP,R5
15079 053214 010546      MOV      R5,-(SP)
15080 053216 016646 000002      MOV      2(SP),-(SP)      : NIB.PTR,*
15081 053222 042716 177770      BIC      #177770,(SP)
15082 053226 012746 000001      MOV      #1,-(SP)
15083 053232 005046      CLR      -(SP)
15084 053234 004767 127700      JSR      PC,BL$GT2
15085 053240 062706 000010      ADD      #10,SP
15086 053244 005700      TST      R0
15087 053246 001011      BNE      25$
15088 053250 011646      MOV      (SP),-(SP)      : NIB.PTR,*
15089 053252 010146      MOV      R1,-(SP)      : NIB.PAT,*
15090 053254 012746 000016      MOV      #16,-(SP)
15091 053260 060616      ADD      SP,(SP)          : ERR.FLG,*
15092 053262 004767 142660      JSR      PC,TST.LNG.WRD
15093 053266 062706 000006      ADD      #6,SP
15094 053272 016605 000010      25$: MOV      10(SP),R5      : ERR.FLG,*
15095 053276 032705 000001      BIT      #1,R5
15096 053302 001012      BNE      26$
15097 053304 021627 000011      CMP      (SP),#11      : NIB.PTR,*
15098 053310 001332      BNE      24$
15099 053312 032705 000001      BIT      #1,R5
15100 053316 001004      BNE      26$
15101 053320 026627 000006 000177      CMP      6(SP),#177      : WRD.CNT,*
15102 053326 001264      BNE      19$
15103 053330 005004      26$: CLR      R4
15104 053332 026627 000002 000003      CMP      2(SP),#3      : PASS.CNT,*
15105 053340 001002      BNE      27$
15106 053342 005204      INC      R4
15107 053344 000403      BR      28$
15108 053346 020527 000001      27$: CMP      R5,#1
15109 053352 001030      BNE      31$
15110 053354 006004      28$: ROR      R4
15111 053356 103010      BCC      29$
15112 053360 005705      TST      R5
15113 053362 001006      BNE      29$
15114 053364 012766 000001 000004      MOV      #1,4(SP)      : *,DONE.FLG
15115 053372 010267 136166      MOV      R2,GOOD.BLK      : SECTOR.NO,*
15116 053376 000405      BR      30$
15117 053400 066702 137520      29$: ADD      RAS,INC,R2
15118 053404 012766 177777 000002      MOV      #-1,2(SP)      : *,PASS.CNT

```

CZPLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 342 E 10

SEQ 0329

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

```

15120      :ML4
15121      :
15122
15123 053412 026627 000004 000001 30$:  CMP      4(SP),#1      ; DONE.FLG,*      7184
15124 053420 001410
15125 053422 016705 136150      BEQ      32$
15126 053426 066705 136130      MOV     LST.ARR,R5
15127 053432 020205      ADD     ARR.INC,R5
15128 053434 001402      CMP     R2,R5      ; SECTOR.NO,*
15129 053436 000167 177040      BEQ     31$:      32$
15130 053442 104467      JMP     28$
15131 053444 006000      TRAP   67
15132 053446 103002      ROR     R0
15133 053450 000167 177010      BHIS   33$:      33$
15134 053454 016705 136116      JMP     18$
15135 053460 066705 136076      MOV     LST.ARR,R5      ;
15136 053464 020205      ADD     ARR.INC,R5      ; SECTOR.NO,*
15137 053466 001026      CMP     R2,R5
15138 053470 104455      BNE     34$:      34$
15139 053472 000117      TRAP   55      ;
15140 053474 010672      .WORD  117
15141 053476 024052      .WORD  INTER
15142 053500 012746 010136      .WORD  DUMPER
15143 053504 012746 007564      MOV     #FNC.15,-(SP)      ;
15144 053510 012746 006442      MOV     #PHR.10,-(SP)
15145 053514 012746 006034      MOV     #WRD.14,-(SP)
15146 053520 012746 000004      MOV     #THR.FMT,-(SP)
15147 053524 010600      MOV     #4,-(SP)
15148 053526 104414      MOV     SP,R0      ; SP,*
15149 053530 016700 140044      TRAP   14
15150 053534 104451      MOV     ML.LUN,R0      ;
15151 053536 104444      TRAP   51
15152 053540 062706 000012      TRAP   44
15153 053544 062706 000014      ADD     #12,SP      ;
15154 053550 000207      ADD     #14,SP      ;
15155      RTS      PC
15156      ; Routine Size: 287 words
15157      ; Maximum stack depth per invocation: 17 words
15162
15163
15167
15171 053552
15172 053552 004767 176676      T29:: 1$:  JSR     PC,$T29
15173 053556 104466      TRAP   66

```

15175
15176
15177
15178 053560 006000
15179 053562 103773
15180 053564 000207
15181
15182
15183
15188
15189
15190 ;

:ML4
:

ROR R0
BLO 1\$
RTS PC

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

7199 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (80)

```

15192 :ML4
15193 :
15194 :
15195 : 7200 :
15196 : 7201 :
15197 : 7202 BGNSTST;
15198 : 7203 :
15199 : 7204 :++
15200 : 7205 : TEST NUMBER: TST 30
15201 : 7206 :
15202 : 7207 : TEST NAME: REFRESH TIMING TEST
15203 : 7208 :
15204 : 7209 : TEST DESCRIPTION:
15205 : 7210 :
15206 : 7211 : TEST THE MEMORY ARRAY MODULES
15207 : 7212 : REFRESH TIMING AND CONTROL LOGIC
15208 : 7213 : REG BY:
15209 : 7214 :
15210 : 7215 : 1. WRITING ALL ONES INTO THE
15211 : 7216 : GOOD BLOCK.
15212 : 7217 :
15213 : 7218 : 2. DELAY FOR 2 MS
15214 : 7219 :
15215 : 7220 : 3. READ THE GOOD BLOCK FOR ONES.
15216 : 7221 : KEEP COUNT OF BAD NIBBLES
15217 : 7222 : ENCOUNTERED
15218 : 7223 :
15219 : 7224 : 4. ALLOW A BAD NIBBLE THRESHOLD
15220 : 7225 : OF 20 BAD NIBBLES OUT OF THE
15221 : 7226 : 1280 NIBBLES TESTED.
15222 : 7227 :
15223 : 7228 : 5. IF THRESHOLD IS EXCEEDED THEN
15224 : 7229 : REPORT ERROR AND DROP THE UNIT.
15225 : 7230 :
15226 : 7231 : IMPLICIT INPUTS:
15227 : 7232 :
15228 : 7233 : PD TEMP:
15229 : 7234 : A BIT VECTOR OF 16 BITS WHERE
15230 : 7235 : THE READ PROM DATA IS STORED
15231 : 7236 : AND ACCESSED FROM.
15232 : 7237 :
15233 : 7238 :
15234 : 7239 :
15235 : 7240 : local
15236 : 7241 : TST_PAT,
15237 : 7242 : ERR_CNT,
15238 : 7243 : ERR_FLG;
15239 : 7244 :
15240 : 7245 : BGNSUB;
15241 : 7246 : CLR_MBUS;
15242 : 7247 : ERR_CNT = ZERO;
15243 : 7248 : TST_PAT = ONES;
15244 : 7249 : MLD1 = .TST_PAT;
15245 : 7250 : MLD2 = .TST_PAT;
15246 : 7251 : MLE2 = .TST_PAT;

```

!TEST PATTERN
!ERROR COUNT
!ERROR FLAG

!LOAD DATA DIAG REGS WITH TST PAT

```

15248 :ML4
15249 :
15250 :
15251 :       7252 DAT_DM_XFER ();
15252 :       7253 MLC51 ≡ write;
15253 :       7254
15254 :       7255 incr WRD_CNT from 0 to 127 do
15255 :       7256   begin
15256 :       7257     DELAY (ONE_US);
15257 :       7258     DAT_CLK = ONE;
15258 :       7259     end;
15259 :       7260
15260 :       7261 CLR_MBUS;
15261 :       7262 DAT_DM_XFER ();
15262 :       7263 MLC51 ≡ read;
15263 :       7264 DELAY (TWO_TH_US);
15264 :       7265
15265 :       7266 incr WRD_CNT from 0 to 127 do
15266 :       7267   begin
15267 :       7268     PD_TEMP = .MLPD;
15268 :       7269     DAT_CLK = ONE;
15269 :       7270     DELAY (ONE_US);
15270 :       7271     RD_LNG_WRD;
15271 :       7272
15272 :       7273   incr NIB_PTR from 0 to 9 do
15273 :       7274     begin
15274 :       7275       if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
15275 :       7276
15276 :       7277       !FIND GOOD NIBBLES AND COMPARE THEM
15277 :       7278
15278 :       7279       if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1; !INCREMENT ERROR COUNT IF ERR_FLG IS SET
15279 :       7280
15280 :       7281     end;
15281 :       7282
15282 :       7283   end;
15283 :       7284     end;
15284 :       7285
15285 :       7286 ENDSUB;
15286 :       7287
15287 :       7288 if .ERR_CNT gtr 20
15288 :       7289 then
15289 :       7290   begin
15290 :       7291     ERRDF (80, ASYNC, DUMPER);
15291 :       7292     PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, FNC_16, WRD_48);
15292 :       7293     DODU (.ML_LDN);
15293 :       7294     DOCLN;
15294 :       7295   end;
15295 :       7296
15296 :       7297 ENDTST;
15300 :

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (80)

!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

!FIND GOOD NIBBLES AND COMPARE THEM

!INCREMENT ERROR COUNT IF ERR_FLG IS SET

!ALLOW 20 NIBBLES TO FAIL

!ERROR IF GTR 20

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

SEQ 0333

15308	053566	004167	130324	\$T30:	JSR	R1,\$SAVE5	:		7198
15309	053572	024646			CMP	-(SP),-(SP)	:		
15310	053574	104402		1\$:	TRAP	2	:		7243
15311	053576	152777	000040	137410	BISB	#40,@ML.REG+40	:		7245
15312	053604	016703	137772		MOV	ML,DUT,R3	:		
15313	053610	042703	177770		BIC	#177770,R3	:		
15314	053614	142777	000007	137372	BICB	#7,@ML.REG+40	:		
15315	053622	150377	137366		BISB	R3,@ML.REG+40	:		
15316	053626	005005			CLR	R5	:	ERR.CNT	7247
15317	053630	012704	177777		MOV	#-1,R4	:	*.TST.PAT	7248
15318	053634	010477	137504		MOV	R4,@ML.REG+170	:	TST.PAT,*	7249
15319	053640	010477	137510		MOV	R4,@ML.REG+200	:	TST.PAT,*	7250
15320	053644	010477	137464		MOV	R4,@ML.REG+160	:	TST.PAT,*	7251
15321	053650	004767	140306		JSR	PC,DAT.DM.XFER	:		7252
15322	053654	012777	000061	137272	MOV	#61,@ML.REG	:		7253
15323	053662	005001			CLR	R1	:	WRD.CNT	7255
15324	053664	012702	000001	2\$:	MOV	#1,R2	:	*,\$\$TMP2	7257
15325	053670	001411		3\$:	BEQ	6\$:		
15326	053672	016703	126220		MOV	LSDLY,R3	:	*,\$\$TMP1	
15327	053676	001404			BEQ	5\$:		
15328	053700	005066	000002	4\$:	CLR	2(SP)	:	\$\$TMP	
15329	053704	005303			DEC	R3	:	\$\$TMP1	
15330	053706	001374			BNE	4\$:		
15331	053710	005302		5\$:	DEC	R2	:	\$\$TMP2	
15332	053712	000766			BR	3\$:		
15333	053714	152777	000020	137352	6\$:	BISB	#20,@ML.REG+120	:	
15334	053722	005201			INC	R1	:	WRD.CNT	7258
15335	053724	020127	000177		CMP	R1,#177	:	WRD.CNT,*	7255
15336	053730	003755			BLE	2\$:		
15337	053732	152777	000040	137254	BISB	#40,@ML.REG+40	:		7259
15338	053740	016703	137636		MOV	ML,DUT,R3	:		
15339	053744	042703	177770		BIC	#177770,R3	:		
15340	053750	142777	000007	137236	BICB	#7,@ML.REG+40	:		
15341	053756	150377	137232		BISB	R3,@ML.REG+40	:		
15342	053762	004767	140174		JSR	PC,DAT.DM.XFER	:		7262
15343	053766	012777	000071	137160	MOV	#71,@ML.REG	:		7263
15344	053774	012702	003720		MOV	#3720,R2	:	*,\$\$TMP2	7264
15345	054000	001411		7\$:	BEQ	10\$:		
15346	054002	016703	126110		MOV	LSDLY,R3	:	*,\$\$TMP1	
15347	054006	001404			BEQ	9\$:		
15348	054010	005066	000002	8\$:	CLR	2(SP)	:	\$\$TMP	
15349	054014	005303			DEC	R3	:	\$\$TMP1	
15350	054016	001374			BNE	8\$:		
15351	054020	005302		9\$:	DEC	R2	:	\$\$TMP2	
15352	054022	000766			BR	7\$:		
15353	054024	005001		10\$:	CLR	R1	:	WRD.CNT	7266
15354	054026	017767	137352	137064	11\$:	MOV	@ML.REG+230,PD.TEMP	:	7268
15355	054034	152777	000020	137232	BISB	#20,@ML.REG+120	:		7269
15356	054042	012702	000001		MOV	#1,R2	:	*,\$\$TMP2	7270
15357	054046	001411		12\$:	BEQ	15\$:		

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

15359          :ML4
15360          :
15361
15362 054050 016703 126042          MOV     LSDLY,R3          ; *,SSTMP1
15363 054054 001404          BEQ     14$
15364 054056 005066 000002 13$: CLR     2(SP)          ; SSTMP
15365 054062 005303          DEC     R3              ; SSTMP1
15366 054064 001374          BNE     13$
15367 054066 005302 14$: DEC     R2              ; SSTMP2
15368 054070 000766          BR      12$
15369 054072 017767 137246 135052 15$: MOV     @ML.REG+170,D1.TEMP
15370 054100 017767 137250 135046 MOV     @ML.REG+200,D2.TEMP
15371 054106 017767 137222 135042 MOV     @ML.REG+160,E2.TEMP
15372 054114 005002          CLR     R2              ; NIB.PTR          7273
15373 054116 010203 16$: MOV     R2,R3          ; NIB.PTR,*      7276
15374 054120 006203          ASR     R3
15375 054122 006203          ASR     R3
15376 054124 006203          ASR     R3
15377 054126 062703 013120          ADD     @PD.TEMP,R3
15378 054132 010346          MOV     R3,-(SP)
15379 054134 010246          MOV     R2,-(SP)          ; NIB.PTR,*
15380 054136 042716 177770          BIC     #177770,(SP)
15381 054142 012746 000001          MOV     #1,-(SP)
15382 054146 005046          CLR     -(SP)
15383 054150 004767 126764          JSR     PC,BLSGT2
15384 054154 062706 000010          ADD     #10,SP
15385 054160 005700          TST     R0
15386 054162 001011          BNE     17$
15387 054164 010246          MOV     R2,-(SP)          ; NIB.PTR,*
15388 054166 010446          MOV     R4,-(SP)          ; TST.PAT,*
15389 054170 012746 000006          MOV     #6,-(SP)
15390 054174 060616          ADD     SP,(SP)          ; ERR.FLG,*
15391 054176 004767 141744          JSR     PC,TST.LNG.WRD
15392 054202 062706 000006          ADD     #6,SP
15393 054206 021627 000001 17$: CMP     (SP),#1          ; ERR.FLG,*          7280
15394 054212 001001          BNE     18$
15395 054214 005205          INC     R5              ; ERR.CNT
15396 054216 005202 18$: INC     R2              ; NIB.PTR          7273
15397 054220 020227 000011          CMP     R2,#11          ; NIB.PTR,*
15398 054224 003734          BLE     16$
15399 054226 005201          INC     R1              ; WRD.CNT          7266
15400 054230 020127 000177          CMP     R1,#177          ; WRD.CNT,*
15401 054234 003674          BLE     11$
15402 054236 104467          TRAP    67              ;
15403 054240 006000          ROR     R0              ;
15404 054242 103002          BHIS   19$
15405 054244 000167 177324          JMP     1$
15406 054250 020527 000024 19$: CMP     R5,#24          ; ERR.CNT,*          7288
15407 054254 003432          BLE     20$
15408 054256 104455          TRAP    55              ;
15409 054260 000120          .WORD  120              ;
15410 054262 010464          .WORD  ASYNC
15411 054264 024052          .WORD  DUMPER
15412 054266 012746 007026          MOV     #WRD.48,-(SP)
15413 054272 012746 010150          MOV     #FNC.16,-(SP)

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

15415      ;ML4
15416      ;
15417      ;
15418 054276 012746 006426      MOV      #WRD.12,-(SP)
15419 054302 012746 007454      MOV      #PHR.4,-(SP)
15420 054306 012746 006530      MOV      #WRD.22,-(SP)
15421 054312 012746 006062      MOV      #FIV.FMT,-(SP)
15422 054316 012746 000006      MOV      #6,-(SP)
15423 054322 010600      MOV      SP,R0      ; SP,*
15424 054324 104414      TRAP     14
15425 054326 016700 137246      MOV      ML.LUN,R0      ;
15426 054332 104451      TRAP     51
15427 054334 104444      TRAP     44
15428 054336 062706 000016      ADD      #16,SP      ;
15429 054342 022626      20$:    CMP      (SP)+,(SP)+      ;
15430 054344 000207      RTS      PC

```

```

; Routine Size: 184 words
; Maximum stack depth per invocation: 15 words

```

```

15431
15432
15433
15438
15439
15443
15447 054346      T30::
15448 054346 004767 177214      1$:    JSR      PC,$T30      ;
15449 054352 104466      TRAP     66
15450 054354 006000      ROR      R0
15451 054356 103773      BLO      1$
15452 054360 000207      RTS      PC

```

```

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

```

```

15453
15454
15455
15460
15461
15462 :          7298 !<BLF/PAGE>

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (81)

```

15464 : PL4
15465 :
15466 :
15467 : 7299 !
15468 : 7300 ! BGNTST;
15469 : 7301 !
15470 : 7302 ! ++
15471 : 7303 ! TEST NUMBER: TST 31
15472 : 7304 !
15473 : 7305 ! TEST NAME: ADDRESS COUNTER TEST
15474 : 7306 !
15475 : 7307 ! TEST DESCRIPTION:
15476 : 7308 ! TEST THE ABILITY OF THE ADDRESS
15477 : 7309 ! COUNTER TO SUCCESSFULLY COUNT
15478 : 7310 ! FROM BLOCK ZERO THROUGH THE
15479 : 7311 ! DEVICES LAST BLOCK BY:
15480 : 7312 !
15481 : 7313 ! 1. WRITING THE LAST BLOCK WITH ONES PATTERN.
15482 : 7314 !
15483 : 7315 ! 2. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK UP TO THE
15484 : 7316 ! LAST BLOCK ADRS MINUS ONE.
15485 : 7317 !
15486 : 7318 ! READ THE LAST BLOCK FOR ONES AND ERROR IF ZEROES.
15487 : 7319 !
15488 : 7320 ! 3. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK THROUGH THE LAST
15489 : 7321 ! BLOCK.
15490 : 7322 !
15491 : 7323 ! READ THE LAST BLOCK FOR ZEROES AND ERROR IF STILL ONES.
15492 : 7324 !
15493 : 7325 ! IMPLICIT INPUTS:
15494 : 7326 ! PD_TEMP:
15495 : 7327 ! A BITVECTOR OF 16 BITS WHERE THE READ FROM DATA IS STORED AND ACCESSED FROM.
15496 : 7328 ! --
15497 : 7329 !
15498 : 7330 ! local
15499 : 7331 ! DODU_FLG, ! DROP UNIT FLAG
15500 : 7332 ! ERR_FLG, ! ERROR FLG
15501 : 7333 ! END_CNT, ! ENDING SECTOR NUMBER
15502 : 7334 ! BG_PAT; ! BACKGROUND PATTERN
15503 : 7335 !
15504 : 7336 ! BGNSUB;
15505 : 7337 ! CLR MBUS;
15506 : 7338 ! DODU_FLG = ZERO;
15507 : 7339 ! BG_PAT = ONES; ! BACKGROUND PAT OF ONES
15508 : 7340 ! MLD1 = .BG_PAT; ! LOAD DATA DIAGS WITH BG PAT
15509 : 7341 ! MLD2 = .BG_PAT;
15510 : 7342 ! MLE2 = .BG_PAT;
15511 : 7343 ! DAT_DM = ONE;
15512 : 7344 ! LAST_BLK_XFER (); ! SET DATA DIAG MODE
15513 : 7345 ! MLCST = write; ! SET UP A LAST BLOCK XFERR
15514 : 7346 ! ! DO A WRITE FUNCTION
15515 : 7347 ! incr WRD_CNT from 0 to 127 do ! WRITE THE LAST BLOCK WITH THE BG PATTERN
15516 : 7348 ! begin
15517 : 7349 ! DELAY (ONE US);
15518 : 7350 ! DAT_CLK = ONE;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
 22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (81)

```

15520 :ML4
15521 :
15522 :
15523 :       7351     end;
15524 :       7352
15525 :       7353     CLR_MBUS;
15526 :       7354     DAT_DM = ONE;
15527 :       7355     LAST_BLK_XFER ();
15528 :       7356     MLCST = read;
15529 :       7357     DELAY (ONE_US);
15530 :       7358
15531 :       7359     incr WD_CNT from 0 to 127 do
15532 :       7360     begin
15533 :       7361     PD_TEMP = .MLPD;
15534 :       7362     DAT_CLK = ONE;
15535 :       7363     DELAY (ONE_US);
15536 :       7364     RD_LNG_WRD;
15537 :       7365
15538 :       7366     incr NIB_PTR from 0 to 9 do
15539 :       7367     begin
15540 :       7368
15541 :       7369     if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .BG_PAT, ERR_FLG);
15542 :       7370
15543 :       7371
15544 :       7372
15545 :       7373     if .ERR_FLG IS_SET
15546 :       7374     then
15547 :       7375     begin
15548 :       7376     ERRDF (81, INTER, DUMPER);
15549 :       7377     PRINTB (THR_FMT, PHR_4, WRD_13, FNC_22);
15550 :       7378     PRINTB (TWO_FMT, FNC_13, WRD_56);
15551 :       7379     EXIT_TST;
15552 :       7380     end;
15553 :       7381
15554 :       7382     end;
15555 :       7383
15556 :       7384     end;
15557 :       7385
15558 :       7386     END_CNT = .LST_BLK - 1;
15559 :       7387     MLD1 = ZEROES;
15560 :       7388     MLD2 = ZEROES;
15561 :       7389     MLE2 = ZEROES;
15562 :       7390
15563 :       7391     incr TWICE from 0 to 1 do
15564 :       7392     begin
15565 :       7393     CLR_MBUS;
15566 :       7394     DAT_DM = ONE;
15567 :       7395     FIRST_BLK_XFER ();
15568 :       7396     MLCST = write;
15569 :       7397
15570 :       7398     incr BLK_CNT from 0 to .END_CNT do
15571 :       7399
15572 :       7400     incr ADRS_CNT from 0 to 127 do
15573 :       7401     begin
15574 :       7402     DAT_CLK = ONE;

```

```

!SET DATA DIAG MODE
!SET UP A LAST BLOCK XFERR
!DO A READ FUNCTION

!READ THE LAST BLOCK FOR BG PATTERN

!GET PROM DATA
!CLOCK OUT THE DATA WORD

!READ THE DATA WORD

!LOOK AT 10 NIBBLES

!FIND GOOD NIBBLES AND COMPARE THEM

!SEE IF ERROR WAS FOUND

!ERROR IF FLG IS SET

!END AT LAST BLOCK -1
!LOAD DATA DIAG REG WITH COMP BG PAT

!REPEAT LOOP TWICE

!SET DATA DIAG MODE
!SET UP A FIRST BLOCK XFERR
!DO A WRITE FUNCTION

!CLOCK THE ADDRESS COUNTER UP TO END_CNT

```

```

15576 :ML4
15577 :
15578 :
15579 : 7403          end;
15580 : 7404
15581 : 7405          CLR_MBUS;
15582 : 7406          DAT_DM = ONE;
15583 : 7407          LAST_BLK_XFER ();
15584 : 7408          MLCST = read;
15585 : 7409          DELAY (ONE_US);
15586 : 7410
15587 : 7411          incr WD_CNT from 0 to 127 do
15588 : 7412          begin
15589 : 7413          PD_TEMP = .MLPD;
15590 : 7414          DAT_CLK = ONE;
15591 : 7415          DELAY (ONE_US);
15592 : 7416          RD_LNG_WRD;
15593 : 7417
15594 : 7418          incr NIB_PTR from 0 to 9 do
15595 : 7419          begin
15596 : 7420
15597 : 7421          if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .9G_PAT, ERR_FLG);
15598 : 7422
15599 : 7423          !FIND GOOD NIBBLES AND COMPARE THEM
15600 : 7424
15601 : 7425          if .ERR_FLG IS_SET
15602 : 7426          then
15603 : 7427          begin
15604 : 7428          ERRDF (82, ASYNC, DUMPER);
15605 : 7429          PRINTB (THR_FMT, WRD_50, WRD_51, WRD_10);
15606 : 7430          DODU_FLG = ONE;
15607 : 7431          end;
15608 : 7432
15609 : 7433          end;
15610 : 7434
15611 : 7435          end;
15612 : 7436
15613 : 7437          END_CNT = .END_CNT + 1;
15614 : 7438          BG_PAT = not .BG_PAT;
15615 : 7439          end;
15616 : 7440
15617 : 7441          ENDSUB;
15618 : 7442
15619 : 7443          if .DODU_FLG IS_SET
15620 : 7444          then
15621 : 7445          begin
15622 : 7446          DODU (.ML_LUN);
15623 : 7447          DOCLN;
15624 : 7448          end;
15625 : 7449
15626 : 7450          ENDTST;
15630 :

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (81)

!SET DATA DIAG MODE
!SET UP A LAST BLOCK XFERR
!DO A READ FUNCTION
!READ THE LAST BLOCK FOR BG PATTERN
!GET THE PROM DATA
!CLOCK OUT DATA WORD
!READ DATA WORD
!LOOK AT 10 NIBBLES
!ERRDF IF FLG IS SET
!NOW END AT THE LAST BLOCK
!COMPLIMENT THE BG PATTERN AND REPEAT
!DROP THIS UNIT IF DODU_FLG IS_SET

15632				:ML4							22-Dec-1980 09:24:31	TOPS
15633				:							22-Dec-1980 09:21:22	PA:<
15634												
15638	054362	004167	127530	\$T31:	JSR	R1,\$\$SAVE5	:					7297
15639	054366	162706	000010		SUB	#10,\$P	:					
15640	054372	104402		1\$:	TRAP	2	:					7334
15641	054374	152777	000040	136612	BISB	#40,@ML.REG+40	:					7336
15642	054402	016705	137174		MOV	ML,DUT,R5	:					
15643	054406	042705	177770		BIC	#177770,R5	:					
15644	054412	142777	000007	136574	BICB	#7,@ML.REG+40	:					
15645	054420	150577	136570		BISB	R5,@ML.REG+40	:					
15646	054424	005066	000002		CLR	2(\$P)	:	DODU.FLG				7338
15647	054430	012701	177777		MOV	#-1,R1	:	*,\$BG.PAT				7339
15648	054434	010177	136704		MOV	R1,@ML.REG+170	:	BG.PAT,*				7340
15649	054440	010177	136710		MOV	R1,@ML.REG+200	:	BG.PAT,*				7341
15650	054444	010177	136664		MOV	R1,@ML.REG+160	:	BG.PAT,*				7342
15651	054450	152777	000010	136616	BISB	#0,@ML.REG+120	:					7343
15652	054456	004767	141432		JSR	PC, LAST.BLK.XFER	:					7344
15653	054462	012777	000061	136464	MOV	#61,@ML.REG	:					7345
15654	054470	005002			CLR	R2	:	WRD.CNT				7347
15655	054472	012703	000001	2\$:	MOV	#1,R3	:	*,\$\$TMP2				7349
15656	054476	001411		3\$:	BEQ	6\$:					
15657	054500	016704	125412		MOV	LSDLY,R4	:	*,\$\$TMP1				
15658	054504	001404			BEQ	5\$:					
15659	054506	005066	000006	4\$:	CLR	6(\$P)	:	\$\$TMP				
15660	054512	005304			DEC	R4	:	\$\$TMP1				
15661	054514	001374			BNE	4\$:					
15662	054516	005303		5\$:	DEC	R3	:	\$\$TMP2				
15663	054520	000766			BR	3\$:					
15664	054522	152777	000020	136544	6\$:	BISB	#20,@ML.REG+120	:				7350
15665	054530	005202			INC	R2	:	WRD.CNT				7347
15666	054532	020227	000177		CMP	R2,#177	:	WRD.CNT,*				
15667	054536	003755			BLE	2\$:					
15668	054540	152777	000040	136446	BISB	#40,@ML.REG+40	:					7351
15669	054546	016705	137030		MOV	ML,DUT,R5	:					
15670	054552	042705	177770		BIC	#177770,R5	:					
15671	054556	142777	000007	136430	BICB	#7,@ML.REG+40	:					
15672	054564	150577	136424		BISB	R5,@ML.REG+40	:					
15673	054570	152777	000010	136476	BISB	#10,@ML.REG+120	:					7354
15674	054576	004767	141312		JSR	PC, LAST.BLK.XFER	:					7355
15675	054602	012777	000071	136344	MOV	#71,@ML.REG	:					7356
15676	054610	012703	000001		MOV	#1,R3	:	*,\$\$TMP2				7357
15677	054614	001411		7\$:	BEQ	10\$:					
15678	054616	016704	125274		MOV	LSDLY,R4	:	*,\$\$TMP1				
15679	054622	001404			BEQ	9\$:					
15680	054624	005066	000006	8\$:	CLR	6(\$P)	:	\$\$TMP				
15681	054630	005304			DEC	R4	:	\$\$TMP1				
15682	054632	001374			BNE	8\$:					
15683	054634	005303		9\$:	DEC	R3	:	\$\$TMP2				
15684	054636	000766			BR	7\$:					
15685	054640	005002		10\$:	CLR	R2	:	WD.CNT				7359

15687						:ML4							
15688						:							
15689						:							
15690	054642	017767	136536	136250	11\$:	MOV	@ML.REG+230,PD.TEMP	:					7361
15691	054650	152777	000020	136416		BISB	#20,@ML.REG+120	:					7362
15692	054656	012703	000001			MOV	#1,R3	:	*,\$\$TMP2				7363
15693	054662	001411			12\$:	BEQ	15\$:					
15694	054664	016704	125226			MOV	LSDLY,R4	:	*,\$\$TMP1				
15695	054670	001404				BEQ	14\$:					
15696	054672	005066	000006		13\$:	CLR	6(SP)	:	\$\$TMP				
15697	054676	005304				DEC	R4	:	\$\$TMP1				
15698	054700	001374				BNE	13\$:					
15699	054702	005303			14\$:	DEC	R3	:	\$\$TMP2				
15700	054704	000766				BR	12\$:					
15701	054706	017767	136432	134236	15\$:	MOV	@ML.REG+170,D1.TEMP	:					
15702	054714	017767	136434	134232		MOV	@ML.REG+200,D2.TEMP	:					
15703	054722	017767	136406	134226		MOV	@ML.REG+160,E2.TEMP	:					
15704	054730	005003				CLR	R3	:	NIB.PTR				7366
15705	054732	010304			16\$:	MOV	R3,R4	:	NIB.PTR,*				7369
15706	054734	006204				ASR	R4	:					
15707	054736	006204				ASR	R4	:					
15708	054740	006204				ASR	R4	:					
15709	054742	062704	013120			ADD	#PD.TEMP,R4	:					
15710	054746	010446				MOV	R4,-(SP)	:					
15711	054750	010346				MOV	R3,-(SP)	:	NIB.PTR,*				
15712	054752	042716	177770			BIC	#177770,(SP)	:					
15713	054756	012746	000001			MOV	#1,-(SP)	:					
15714	054762	005046				CLR	-(SP)	:					
15715	054764	004767	126150			JSR	PC,BL\$GT2	:					
15716	054770	062706	000010			ADD	#10,SP	:					
15717	054774	005700				TST	R0	:					
15718	054776	001011				BNE	17\$:					
15719	055000	010346				MOV	R3,-(SP)	:	NIB.PTR,*				
15720	055002	010146				MOV	R1,-(SP)	:	BG.PAT,*				
15721	055004	012746	000012			MOV	#12,-(SP)	:					
15722	055010	060616				ADD	SP,(SP)	:	ERR.FLG,*				
15723	055012	004767	141130			JSR	PC,TST.LNG.WRD	:					
15724	055016	062706	000006			ADD	#6,SP	:					
15725	055022	026627	000004	000001	17\$:	CMP	4(SP),#1	:	ERR.FLG,*				7373
15726	055030	001037				BNE	18\$:					
15727	055032	104455				TRAP	55	:					7376
15728	055034	000121				.WORD	121	:					
15729	055036	010672				.WORD	INTER	:					
15730	055040	024052				.WORD	DUMPER	:					
15731	055042	012746	010230			MOV	#FNC.22,-(SP)	:					7377
15732	055046	012746	006436			MOV	#WRD.13,-(SP)	:					
15733	055052	012746	007454			MOV	#PHR.4,-(SP)	:					
15734	055056	012746	006034			MOV	#THR.FMT,-(SP)	:					
15735	055062	012746	000004			MOV	#4,-(SP)	:					
15736	055066	010600				MOV	SP,R0	:	SP,*				
15737	055070	104414				TRAP	14	:					
15738	055072	012716	007122			MOV	#WRD.56,(SP)	:					7378
15739	055076	012746	010112			MOV	#FNC.13,-(SP)	:					
15740	055102	012746	006024			MOV	#TWO.FMT,-(SP)	:					
15741	055106	012746	000003			MOV	#3,-(SP)	:					

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Line No.
15799						:ML4		
15800						:		
15801								
15802	055376	005002				26\$: CLR R2	: WD.CNT	7411
15803	055400	017767	136000	135512		27\$: MOV @ML.REG+230,PD.TEMP	:	7413
15804	055406	152777	000020	135660		BISB #20,@ML.REG+120	:	7414
15805	055414	012703	000001			MOV #1,R3	: *,SSTMP2	7415
15806	055420	001411				28\$: BEQ 31\$:	
15807	055422	016704	124470			MOV L\$DLY,R4	: *,SSTMP1	
15808	055426	001404				BEQ 30\$:	
15809	055430	005066	000006			29\$: CLR 6(SP)	: SSTMP	
15810	055434	005304				DEC R4	: SSTMP1	
15811	055436	001374				BNE 29\$:	
15812	055440	005303				30\$: DEC R3	: SSTMP2	
15813	055442	000766				BR 28\$:	
15814	055444	017767	135674	133500		31\$: MOV @ML.REG+170,D1.TEMP		
15815	055452	017767	135676	133474		MOV @ML.REG+200,D2.TEMP		
15816	055460	017767	135650	133470		MOV @ML.REG+160,E2.TEMP		
15817	055466	005003				CLR R3	: NIB.PTR	7418
15818	055470	010304				32\$: MOV R3,R4	: NIB.PTR,*	7421
15819	055472	006204				ASR R4		
15820	055474	006204				ASR R4		
15821	055476	006204				ASR R4		
15822	055500	062704	013120			ADD #PD.TEMP,R4		
15823	055504	010446				MOV R4,-(SP)		
15824	055506	010346				MOV R3,-(SP)	: NIB.PTR,*	
15825	055510	042716	177770			BIC #177770,(SP)		
15826	055514	012746	000001			MOV #1,-(SP)		
15827	055520	005046				CLR -(SP)		
15828	055522	004767	125412			JSR PC,BL\$GT2		
15829	055526	062706	000010			ADD #10,SP		
15830	055532	005700				TST R0		
15831	055534	001011				BNE 33\$		
15832	055536	010346				MOV R3,-(SP)	: NIB.PTR,*	
15833	055540	010146				MOV R1,-(SP)	: BG.PAT,*	
15834	055542	012746	000012			MOV #12,-(SP)		
15835	055546	060616				ADD SP,(SP)	: ERR.FLG,*	
15836	055550	004767	140372			JSR PC,TST.LNG.WRD		
15837	055554	062706	000006			ADD #6,SP		
15838	055560	026627	000004	000001		33\$: CMP 4(SP),#1	: ERR.FLG,*	7425
15839	055566	001025				BNE 34\$		
15840	055570	104455				TRAP 55	:	7428
15841	055572	000122				.WORD 122		
15842	055574	010464				.WORD ASYNC		
15843	055576	024052				.WORD DUMPER		
15844	055600	012746	006406			MOV #WRD.10,-(SP)	:	7429
15845	055604	012746	007054			MOV #WRD.51,-(SP)		
15846	055610	012746	007046			MOV #WRD.50,-(SP)		
15847	055614	012746	006034			MOV #THR.FMT,-(SP)		
15848	055620	012746	000004			MOV #4,-(SP)		
15849	055624	010600				MOV SP,R0	: SP,*	
15850	055626	104414				TRAP 14		
15851	055630	012766	000001	000014		MOV #1,14(SP)	: *,DODU.FLG	7430
15852	055636	062706	000012			ADD #12,SP	:	7427
15853	055642	005203				34\$: INC R3	: NIB.PTR	7418

```

15855      ;ML4
15856      ;
15857
15858 055644 020327 000011      CMP      R3,#1:      ; NIB.PTR,*
15859 055650 003707      BLE      32$
15860 055652 005202      INC      R2      ; WD.CNT      7411
15861 055654 020227 000177      CMP      R2,#177      ; WD.CNT,*
15862 055660 003647      BLE      27$
15863 055662 005216      INC      (SP)      ; END.CNT      7437
15864 055664 005101      COM      R1      ; BG.PAT      7438
15865 055666 005205      INC      R5      ; TWICE      7391
15866 055670 020527 000001      CMP      R5,#1      ; TWICE,*
15867 055674 003002      BGT      35$
15868 055676 000167 177272      JMP      19$
15869 055702 104467      35$: TRAP      67      ;      7439
15870 055704 006000      ROR      R0
15871 055706 103002      BHIS     36$
15872 055710 000167 176456      JMP      1$
15873 055714 026627 000002 000001 36$: CMP      2(SP),#1      ; DODU.FLG,*      7443
15874 055722 001004      BNE      37$
15875 055724 016700 135650      MOV      ML,LUN,R0      ;      7446
15876 055730 104451      TRAP     51
15877 055732 104444      TRAP     44
15878 055734 062706 000010      37$: ADD      #10,SP      ;      7297
15879 055740 000207      RTS      PC
15880
15881      ; Routine Size: 376 words
15882      ; Maximum stack depth per invocation: 18 words
15887
15888
15892
15896 055742      T31::
15897 055742 004767 176414      1$: JSR      PC,$T31      ;      7448
15898 055746 104466      TRAP     66
15899 055750 006000      ROR      R0
15900 055752 103773      BLO      1$
15901 055754 000207      RTS      PC
15902
15903      ; Routine Size: 6 words
15904      ; Maximum stack depth per invocation: 0 words

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 357^{G 11}

SEQ 0344

15910 ;ML4
15911 :
15912 :
15913 : 7451 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (81)

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (82)

```

15915 :ML4
15916 :
15917 :
15918 : 7452 |
15919 : 7453 |
15920 : 7454 | BGNSTST:
15921 : 7455 |
15922 : 7456 | ++
15923 : 7457 | TEST NUMBER: TST 32
15924 : 7458 |
15925 : 7459 | TEST NAME: ARRAY MODULE SELECTION TEST
15926 : 7460 |
15927 : 7461 | TEST DESCRIPTION:
15928 : 7462 |
15929 : 7463 | TEST FOR UNIQUE ARRAY MODULE
15930 : 7464 | SELECTION BY:
15931 : 7465 |
15932 : 7466 | 1. WRITING THE RESPECTIVE ARRAY
15933 : 7467 | MODULE POSITION NUMBER INTO
15934 : 7468 | THE FIRST GOOD NIBBLE FOUND
15935 : 7469 | IN THE ARRAY. DO FOR ALL
15936 : 7470 | PRESENT ARRAYS.
15937 : 7471 |
15938 : 7472 | 2. READ THE ARRAYS FOR THEIR
15939 : 7473 | RESPECTIVE POSTION NUMBERS.
15940 : 7474 |
15941 : 7475 | IMPLICIT INPUTS:
15942 : 7476 | PD TEMP:
15943 : 7477 | A BIT VECTOR OF 16 BITS WHERE
15944 : 7478 | THE READ FROM DATA IS STORED
15945 : 7479 | AND ACCESSED FROM.
15946 : 7480 |
15947 : 7481 | IO BUF :
15948 : 7482 | A VECTOR OF 256 WORDS WHERE
15949 : 7483 | DATA FOR MBUS READS AND WRITE
15950 : 7484 | FUNCTION ARE FOUND.
15951 : 7485 |
15952 : 7486 |
15953 : 7487 |
15954 : 7488 | Local
15955 : 7489 | WRDS_TSTED,
15956 : 7490 | ARR_SEL,
15957 : 7491 | FND_GD_NIB,
15958 : 7492 | ARR_NUM;
15959 : 7493 |
15960 : 7494 | ARR_SEL = ZEROES;
15961 : 7495 | ARR_SEL = .ARR_SEL - .ARR_INC;
15962 : 7496 |
15963 : 7497 | incr ARR_CNT from 0 to .OP_NUM_ARR do
15964 : 7498 | begin
15965 : 7499 | CLR_MBUS;
15966 : 7500 | FND_GD_NIB = ZERO;
15967 : 7501 | WRDS_TSTED = ZERO;
15968 : 7502 | DAT_DM = ONE;
15969 : 7503 | MLWC = not 255;

```

```

!WRDS TESTED
!ARRAY SELECT
!FOUND GOOD NIBBLE
!ARRAY NUMBER

```

```
!START ARR_SEL BACK ONE ARRAY
```

```
!TEST ALL PRESENT ARRAYS
```

```
!SET DATA DIAG MODE
!LOAD WORD COUNT
```

```

15971 :ML4
15972 :
15973 :
15974 : 7504 MLBA = IO_BUF;
15975 : 7505 ARR_SEL = .ARR_SEL + .ARR_INC;
15976 : 7506 MLD1 = .ARR_SEL;
15977 : 7507 MLCS1 = write;
15978 : 7508
15979 : 7509 do
15980 : 7510 begin
15981 : 7511 DELAY (ONE_US);
15982 : 7512 PD_TEMP = .MLPD;
15983 : 7513 WRDS_TSTED = .WRDS_TSTED + 1;
15984 : 7514
15985 : 7515 incr CNT from 0 to 8 do
15986 : 7516
15987 : 7517     if .PD_TEMP [.CNT] IS_NOT_SET
15988 : 7518     then
15989 : 7519         begin
15990 : 7520
15991 : 7521             case .CNT from 0 to 8 of
15992 : 7522                 set
15993 : 7523                 [0] :
15994 : 7524                     (MLD1)<0, 4> = .ARR_CNT;
15995 : 7525
15996 : 7526                 [1] :
15997 : 7527                     (MLD1)<4, 4> = .ARR_CNT;
15998 : 7528
15999 : 7529                 [2] :
16000 : 7530                     (MLD1)<8, 4> = .ARR_CNT;
16001 : 7531
16002 : 7532                 [3] :
16003 : 7533                     (MLD1)<12, 4> = .ARR_CNT;
16004 : 7534
16005 : 7535                 [4] :
16006 : 7536                     (MLD2)<0, 4> = .ARR_CNT;
16007 : 7537
16008 : 7538                 [5] :
16009 : 7539                     (MLD2)<4, 4> = .ARR_CNT;
16010 : 7540
16011 : 7541                 [6] :
16012 : 7542                     (MLD2)<8, 4> = .ARR_CNT;
16013 : 7543
16014 : 7544                 [7] :
16015 : 7545                     (MLD2)<12, 4> = .ARR_CNT;
16016 : 7546
16017 : 7547                 [8] :
16018 : 7548                     (MLE2)<8, 4> = .ARR_CNT;
16019 : 7549
16020 : 7550             tes;
16021 : 7551
16022 : 7552             FND_GD_NIB = ONE;
16023 : 7553             exitloop;
16024 : 7554             end;
16025 : 7555

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (82)

!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

!SET FOUND GOOD NIBBLE FLG
!EXIT THE LOOP

```

16027 :ML4
16028 :
16029 :
16030 : 7556 DAT_CLK = ONE; !CLOCK GOOD NIBBLE INTO MEMORY & GET NXT FROM WRD
16031 : 7557 end
16032 : 7558 until (.FND_GD_NIB IS_SET ) or (.WRDS_TSTED eql 14); !DO UNTIL FOUND GOOD NIBBLE OR 14 WRDS TSTED
16033 : 7559
16034 : 7560 if .WRDS_TSTED eql 14 !IF 14 WORDS TSTED
16035 : 7561 then
16036 : 7562 begin !THEN ERROR AND EXIT TESTED
16037 : 7563 ERRDF (111, INTER, DUMPER);
16038 : 7564 PRINTB (SEV_FMT, WRD_14, PHR_10, FNC_15, WRD_12, FNC_17, WRD_37, WRD_56);
16039 : 7565 EXIT_TST;
16040 : 7566 end;
16041 : 7567
16042 : 7568 end;
16043 : 7569
16044 : 7570 ARR_SEL = ZEROES;
16045 : 7571 ARR_SEL = .ARR_SEL - .ARR_INC; !START ARR SEL BACK ONE AGAIN
16046 : 7572
16047 : 7573 incr ARR_CNT from 0 to .OP_NUM_ARR do !TEST ALL PRESENT ARRAYS
16048 : 7574 begin
16049 : 7575 BGNSUB;
16050 : 7576 CLR_MBUS;
16051 : 7577 FND_GD_NIB = ZERO;
16052 : 7578 DAT_DM = ONE; !SET DATA DIAG MODE
16053 : 7579 MLWC = not 255; !LOAD WORD COUNT
16054 : 7580 MLBA = IO_BUF; !LOAD MBUS ADDRESS
16055 : 7581 ARR_SEL = .ARR_SEL + .ARR_INC; !LOOK AT NEXT ARRAY
16056 : 7582 MLDA = .ARR_SEL; !LOAD DSA WITH ARRAY SELECT
16057 : 7583 MLCS1 = read; !DO A READ FUNCTION
16058 : 7584
16059 : 7585 do !THIS LOOP READS ARRAYS FOR ARRAY NUMBERS
16060 : 7586 begin
16061 : 7587 DELAY (ONE_US);
16062 : 7588 PD_TEMP = .MLPD; !GET THE PROM DATA
16063 : 7589
16064 : 7590 incr CNT from 0 to 8 do !LOOK AT 9 NIBBLES
16065 : 7591
16066 : 7592 if .PD_TEMP [.CNT] IS_NOT_SET !FIND THE GOOD NIBBLES WHERE ARR NUM IS STORED
16067 : 7593 then
16068 : 7594 begin
16069 : 7595 DAT_CLK = ONE; !CLOCK ARRAY WORD OUT
16070 : 7596
16071 : 7597 case .CNT from 0 to 8 of !SELECT AND READ GOOD NIBBLE
16072 : 7598 set
16073 : 7599 [0] :
16074 : 7600 ARR_NUM = .(MLD1)<0, 4>; !NIBBLE 0
16075 : 7601
16076 : 7602 [1] :
16077 : 7603 ARR_NUM = .(MLD1)<4, 4>; !NIBBLE 1
16078 : 7604
16079 : 7605 [2] :
16080 : 7606 ARR_NUM = .(MLD1)<8, 4>; !NIBBLE 2
16081 : 7607

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (82)

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (82)

```

16083 :ML4
16084 :
16085 :
16086 : 7608
16087 : 7609 [3] :
16088 : 7610 ARR_NUM = .(MLD1)<12, 4>; !NIBBLE 3
16089 : 7611
16090 : 7612 [4] :
16091 : 7613 ARR_NUM = .(MLD2)<0, 4>; !NIBBLE 4
16092 : 7614
16093 : 7615 [5] :
16094 : 7616 ARR_NUM = .(MLD2)<4, 4>; !NIBBLE 5
16095 : 7617
16096 : 7618 [6] :
16097 : 7619 ARR_NUM = .(MLD2)<8, 4>; !NIBBLE 6
16098 : 7620
16099 : 7621 [7] :
16100 : 7622 ARR_NUM = .(MLD2)<12, 4>; !NIBBLE 7
16101 : 7623
16102 : 7624 [8] :
16103 : 7625 ARR_NUM = .(MLE2)<8, 4>; !NIBBLE 8
16104 : 7626 tes;
16105 : 7627
16106 : 7628 FND_GD_NIB = ONE; !SET FND GD NIB FLG
16107 : 7629 exitloop; !EXIT LOOP
16108 : 7630 end;
16109 : 7631
16110 : 7632 DAT_CLK = ONE; !CLOCK OUT NEXT PROM LOCATION
16111 : 7633 end
16112 : 7634 until .FND_GD_NIB IS_SET; !REPEAT UNTIL FOUND THE GOOD NIBBLE
16113 : 7635
16114 : 7636 if .ARR_CNT neq .ARR_NUM !SEE IF ARRAY CONTAINS IT'S ARR NUM
16115 : 7637 then
16116 : 7638 begin !ERROR IF NOT THERE
16117 : 7639 ERRDF (83, ASYNC, DUMPER);
16118 : 7640 PRINTB (FOR_FMT, WRD_39, FNC 17, WRD_37, WRD_10);
16119 : 7641 PRINTB (FMT_14, .ARR_CNT, .ARR_NUM);
16120 : 7642 end;
16121 : 7643
16122 : 7644 ENDSUB;
16123 : 7645 end;
16124 : 7646
16125 : 7647 ENDTST;
16129 :
16133 055756 004167 126134 $T32: JSR R1,$$SAVE5 ; 7450
16134 055762 162706 000006 SUB #6,SP ;
16135 055766 005005 CLR R5 ; ARR.SEL 7494
16136 055770 166705 133566 SUB ARR.INC,R5 ; *.ARR.SEL 7495

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

Address	Op-Code	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Address
16138								
16139								
16140								
16141	055774	016766	133560	000002		MOV	OP.NUM.ARR,2(SP)	
16142	056002	005003				CLR	R3	: ARR.CNT
16143	056004	000167	000702			JMP	27\$	
16144	056010	152777	000040	135176	1\$:	BISB	#40,@ML.REG+40	
16145	056016	016702	135560			MOV	ML,DUT,R2	
16146	056022	042702	177770			BIC	#177770,R2	
16147	056026	142777	000007	135160		BICB	#7,@ML.REG+40	
16148	056034	150277	135154			BISB	R2,@ML.REG+40	
16149	056040	005016				CLR	(SP)	: FND.GD.NIB
16150	056042	005004				CLR	R4	: WRDS.TSTED
16151	056044	152777	000010	135222		BISB	#10,@ML.REG+120	
16152	056052	012777	177400	135104		MOV	#-400,@ML.REG+10	
16153	056060	012777	011600	135106		MOV	#10.BUF,@ML.REG+20	
16154	056066	066705	133470			ADD	ARR.INC,R5	: *,ARR.SEL
16155	056072	010577	135106			MOV	R5,@ML.REG+30	: ARR.SEL,*
16156	056076	012777	000061	135050		MOV	#61,@ML.REG	
16157	056104	012701	000001		2\$:	MOV	#1,R1	: *,SSTMP2
16158	056110	001411			3\$:	BEQ	6\$	
16159	056112	016702	124000			MOV	LDLY,R2	: *,SSTMP1
16160	056116	001404				BEQ	5\$	
16161	056120	005066	000004		4\$:	CLR	4(SP)	: SSTMP
16162	056124	005302				DEC	R2	: SSTMP1
16163	056126	001374				BNE	4\$	
16164	056130	005301			5\$:	DEC	R1	: SSTMP2
16165	056132	000766				BR	3\$	
16166	056134	017767	135244	134756	6\$:	MOV	@ML.REG+230,PD.TEMP	
16167	056142	005204				INC	R4	: WRDS.TSTED
16168	056144	005002				CLR	R2	: CNT
16169	056146	010201			7\$:	MOV	R2,R1	: CNT,*
16170	056150	006201				ASR	R1	
16171	056152	006201				ASR	R1	
16172	056154	006201				ASR	R1	
16173	056156	062701	013120			ADD	#PD.TEMP,R1	
16174	056162	010146				MOV	R1,-(SP)	
16175	056164	010246				MOV	R2,-(SP)	: CNT,*
16176	056166	042716	177770			BIC	#177770,(SP)	
16177	056172	012746	000001			MOV	#1,-(SP)	
16178	056176	005046				CLR	-(SP)	
16179	056200	004767	124734			JSR	PC,BLSGT2	
16180	056204	062706	000010			ADD	#10,SP	
16181	056210	005700				TST	R0	
16182	056212	001155				BNE	23\$	
16183	056214	010201				MOV	R2,R1	: CNT,*
16184	056216	006301				ASL	R1	
16185	056220	066107	056224			ADD	8\$(R1),PC	
16186	056224	000022			8\$:	.WORD	9\$-8\$	
16187	056226	000040				.WORD	10\$-8\$	
16188	056230	000072				.WORD	12\$-8\$	
16189	056232	000112				.WORD	13\$-8\$	
16190	056234	000146				.WORD	15\$-8\$	
16191	056236	000164				.WORD	16\$-8\$	
16192	056240	000216				.WORD	18\$-8\$	7521


```

16194      :ML4
16195      :
16196
16197 056242 000236      .WORD 19S-8S
16198 056244 000272      .WORD 21S-8S
16199 056246 010301      9S:  MOV R3,R1      ; ARR.CNT,* 7525
16200 056250 042701 177760  BIC #177760,R1
16201 056254 142777 000017 135062  BICB #17,@ML.REG+170
16202 056262 000412      BR 11S
16203 056264 010301      10S: MOV R3,R1      ; ARR.CNT,* 7528
16204 056266 006301      ASL R1
16205 056270 006301      ASL R1
16206 056272 006301      ASL R1
16207 056274 006301      ASL R1
16208 056276 042701 177417  BIC #177417,R1
16209 056302 142777 000360 135034  BICB #360,@ML.REG+170
16210 056310 150177 135030 11S: BISB R1,@ML.REG+170
16211 056314 000511      BR 22S
16212 056316 010301      12S: MOV R3,R1      ; ARR.CNT,* 7521
16213 056320 000301      ; ARR.CNT,* 7531
16214 056322 042701 170377  BIC #170377,R1
16215 056326 042777 007400 135010  BIC #7400,@ML.REG+170
16216 056334 000413      BR 14S
16217 056336 010301      13S: MOV R3,R1      ; ARR.CNT,* 7534
16218 056340 000301      SWAB R1
16219 056342 006301      ASL R1
16220 056344 006301      ASL R1
16221 056346 006301      ASL R1
16222 056350 006301      ASL R1
16223 056352 042701 007777  BIC #7777,R1
16224 056356 042777 170000 134760  BIC #170000,@ML.REG+170
16225 056364 050177 134754 14S: BIS R1,@ML.REG+170
16226 056370 000463      BR 22S
16227 056372 010301      15S: MOV R3,R1      ; ARR.CNT,* 7521
16228 056374 042701 177760  BIC #177760,R1      ; ARR.CNT,* 7537
16229 056400 142777 000017 134746  BICB #17,@ML.REG+200
16230 056406 000412      BR 17S
16231 056410 010301      16S: MOV R3,R1      ; ARR.CNT,* 7540
16232 056412 006301      ASL R1
16233 056414 006301      ASL R1
16234 056416 006301      ASL R1
16235 056420 006301      ASL R1
16236 056422 042701 177417  BIC #177417,R1
16237 056426 142777 000360 134720  BICB #360,@ML.REG+200
16238 056434 150177 134714 17S: BISB R1,@ML.REG+200
16239 056440 000437      BR 22S
16240 056442 010301      18S: MOV R3,R1      ; ARR.CNT,* 7521
16241 056444 000301      ; ARR.CNT,* 7543
16242 056446 042701 170377  BIC #170377,R1
16243 056452 042777 007400 134674  BIC #7400,@ML.REG+200
16244 056460 000413      BR 20S
16245 056462 010301      19S: MOV R3,R1      ; ARR.CNT,* 7546
16246 056464 000301      SWAB R1
16247 056466 006301      ASL R1
16248 056470 006301      ASL R1

```

16250									
16251									
16252									
16253	056472	006301							
16254	056474	006301							
16255	056476	042701	007777						
16256	056502	042777	170000	134644					
16257	056510	050177	134640		20\$:				
16258	056514	000411							7521
16259	056516	010301			21\$:				7549
16260	056520	000301							
16261	056522	042701	170377						
16262	056526	042777	007400	134600					
16263	056534	050177	134574						
16264	056540	012716	000001		22\$:				7552
16265	056544	000406							7553
16266	056546	005202			23\$:				7515
16267	056550	020227	000010						
16268	056554	003002							
16269	056556	000167	177364						
16270	056562	152777	000020	134504	24\$:				7556
16271	056570	021627	000001						7558
16272	056574	001405							
16273	056576	020427	000016						
16274	056602	001402							
16275	056604	000167	177274						
16276	056610	020427	000016		25\$:				7560
16277	056614	001035							
16278	056616	104455							7563
16279	056620	000157							
16280	056622	010672							
16281	056624	024052							
16282	056626	012746	007122						7564
16283	056632	012746	006700						
16284	056636	012746	010162						
16285	056642	012746	006426						
16286	056646	012746	010136						
16287	056652	012746	007564						
16288	056656	012746	006442						
16289	056662	012746	006120						
16290	056666	012746	000010						
16291	056672	010600							
16292	056674	104414							
16293	056676	104463							
16294	056700	062706	000022						7560
16295	056704	000167	000562						7562
16296	056710	005203			26\$:				7497
16297	056712	020366	000002		27\$:				
16298	056716	003002							
16299	056720	000167	177064						
16300	056724	005005			28\$:				7570
16301	056726	166705	132630						7571
16302	056732	016766	132622	000002					7573
16303	056740	005004							
16304	056742	000167	000516						

Line No	Address	Code	Label	Machine Code	Assembly Code	Comments	Seq No
16306			:ML4				
16307			:				
16308							
16309	056746	104402	29\$: TRAP	2			7574
16310	056750	152777	134236 BISB	#40, @ML.REG+40			7575
16311	056756	016702	MOV	ML, DUT, R2			
16312	056762	042702	BIC	#177770, R2			
16313	056766	142777	134220 BICB	#7, @ML.REG+40			
16314	056774	150277	BISB	R2, @ML.REG+40			
16315	057000	005016	CLR	(SP)	: FND.GD.NIB		7577
16316	057002	152777	134264 BISB	#10, @ML.REG+120			7578
16317	057010	012777	134146 MOV	#-400, @ML.REG+10			7579
16318	057016	012777	134150 MOV	#10, BUF, @ML.REG+20			7580
16319	057024	066705	ADD	ARR, INC, R5	: *,ARR.SEL		7581
16320	057030	010577	MOV	R5, @ML.REG+30	: ARR.SEL,*		7582
16321	057034	012777	134112 MOV	#71, @ML.REG			7583
16322	057042	012701	000001 30\$: MOV	#1, R1	: *,SSTMP2		7587
16323	057046	001411	31\$: BEQ	34\$			
16324	057050	016702	MOV	LSDLY, R2	: *,SSTMP1		
16325	057054	001404	BEQ	33\$			
16326	057056	005066	000004 32\$: CLR	4(SP)	: SSTMP		
16327	057062	005302	DEC	R2	: SSTMP1		
16328	057064	001374	BNE	32\$			
16329	057066	005301	33\$: DEC	R1	: SSTMP2		
16330	057070	000766	BR	31\$			
16331	057072	017767	134306 134020 34\$: MOV	@ML.REG+230, PD.TEMP			7588
16332	057100	005002	CLR	R2	: CNT		7590
16333	057102	010201	35\$: MOV	R2, R1	: CNT,*		7592
16334	057104	006201	ASR	R1			
16335	057106	006201	ASR	R1			
16336	057110	006201	ASR	R1			
16337	057112	062701	013120 ADD	#PD.TEMP, R1			
16338	057116	010146	MOV	R1, -(SP)			
16339	057120	010246	MOV	R2, -(SP)	: CNT,*		
16340	057122	042716	177770 BIC	#177770, (SP)			
16341	057126	012746	000001 MOV	#1, -(SP)			
16342	057132	005046	CLR	-(SP)			
16343	057134	004767	124000 JSR	PC, BL\$GT2			
16344	057140	062706	000010 ADD	#10, SP			
16345	057144	005700	TST	R0			
16346	057146	001070	BNE	50\$			
16347	057150	152777	000020 134116 BISB	#20, @ML.REG+120			7595
16348	057156	010201	MOV	R2, R1	: CNT,*		7597
16349	057160	006301	ASL	R1			
16350	057162	066107	057166 ADD	36\$(R1), PC			
16351	057166	000022	36\$: .WORD	37\$-36\$			
16352	057170	000030	.WORD	38\$-36\$			
16353	057172	000036	.WORD	39\$-36\$			
16354	057174	000044	.WORD	40\$-36\$			
16355	057176	000052	.WORD	41\$-36\$			
16356	057200	000060	.WORD	42\$-36\$			
16357	057202	000076	.WORD	44\$-36\$			
16358	057204	000104	.WORD	45\$-36\$			
16359	057206	000122	.WORD	47\$-36\$			
16360	057210	117703	134130 37\$: MOVB	@ML.REG+170, R3	: *,ARR.NUM		7601

```

16362      ;ML4
16363      ;
16364
16365 057214 000440      BR      49$
16366 057216 117703 134122 38$: MOVB  @ML.REG+170,R3      ; *,ARR.NUM      7604
16367 057222 000413      BR      43$
16368 057224 017703 134114 39$: MOV   @ML.REG+170,R3      ; *,ARR.NUM      7607
16369 057230 000431      BR      48$
16370 057232 017703 134106 40$: MOV   @ML.REG+170,R3      ; *,ARR.NUM      7610
16371 057236 000417      BR      46$
16372 057240 117703 134110 41$: MOVB  @ML.REG+200,R3     ; *,ARR.NUM      7613
16373 057244 000424      BR      49$
16374 057246 117703 134102 42$: MOVB  @ML.REG+200,R3     ; *,ARR.NUM      7616
16375 057252 006203      43$: ASR   R3                ; ARR.NUM
16376 057254 006203      ASR   R3                ; ARR.NUM
16377 057256 006203      ASR   R3                ; ARR.NUM
16378 057260 006203      ASR   R3                ; ARR.NUM
16379 057262 000415      BR      49$
16380 057264 017703 134064 44$: MOV   @ML.REG+200,R3     ; *,ARR.NUM      7619
16381 057270 000411      BR      48$
16382 057272 017703 134056 45$: MOV   @ML.REG+200,R3     ; *,ARR.NUM      7622
16383 057276 006203      46$: ASR   R3                ; ARR.NUM
16384 057300 006203      ASR   R3                ; ARR.NUM
16385 057302 006203      ASR   R3                ; ARR.NUM
16386 057304 006203      ASR   R3                ; ARR.NUM
16387 057306 000402      BR      48$
16388 057310 017703 134020 47$: MOV   @ML.REG+160,R3     ; *,ARR.NUM      7625
16389 057314 000303      48$: SWAB  R3                ; ARR.NUM
16390 057316 042703 177760 49$: BIC   #177760,R3        ; *,ARR.NUM
16391 057322 012716 000001      MOV   #1,(SP)           ; *,FND.GD.NIB      7628
16392 057326 000404      BR      51$
16393 057330 005202      50$: INC   R2                ; CNT                7629
16394 057332 020227 000010      CMP   R2,#10           ; CNT,*              7590
16395 057336 003661      BLE   35$
16396 057340 152777 000020 133726 51$: BLSB  #20,@ML.REG+120   ;
16397 057346 021627 000001      CMP   (SP),#1          ; FND.GD.NIB,*      7632
16398 057352 001233      BNE   30$              ;
16399 057354 020403      CMP   R4,R3            ; ARR.CNT,ARR.NUM   7636
16400 057356 001434      BEQ   52$              ;
16401 057360 104455      TRAP  55                ;
16402 057362 000123      .WORD 123                ;
16403 057364 010464      .WORD ASYNC              ;
16404 057366 024052      .WORD DUMPER            ;
16405 057370 012746 006406      MOV   #WRD.10,-(SP)     ;
16406 057374 012746 006700      MOV   #WRD.37,-(SP)     ;
16407 057400 012746 010162      MOV   #FNC.17,-(SP)    ;
16408 057404 012746 006716      MOV   #WRD.39,-(SP)    ;
16409 057410 012746 006046      MOV   #FOR.FMT,-(SP)   ;
16410 057414 012746 000005      MOV   #5,-(SP)         ;
16411 057420 010600      MOV   SP,R0            ; SP,*
16412 057422 104414      TRAP  14                ;
16413 057424 010316      MOV   R3,(SP)          ; ARR.NUM,*
16414 057426 010446      MOV   R4,-(SP)         ; ARR.CNT,*
16415 057430 012746 005110      MOV   #FMT.14,-(SP)    ;
16416 057434 012746 000003      MOV   #3,-(SP)
    
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

7604
7607
7610
7613
7616
7619
7622
7625
7628
7629
7590
7632
7634
7636
7639
7640
7641

```
16418 ;ML4
16419 ;
16420 ;
16421 057440 010600 MOV SP,R0 ; SP,*
16422 057442 104414 TRAP 14
16423 057444 062706 000022 ADD #22,SP ;
16424 057450 104467 52$: TRAP 67 ;
16425 057452 006000 ROR R0 ;
16426 057454 103002 BHIS 54$
16427 057456 000167 177264 53$: JMP 29$
16428 057462 005204 54$: INC R4 ; ARR.CNT
16429 057464 020466 000002 55$: CMP R4,2(SP) ; ARR.CNT,*
16430 057470 003772 BLE 53$
16431 057472 062706 000006 56$: ADD #6,SP ;
16432 057476 000207 RTS PC ;
16433 ;
16434 ; Routine Size: 425 words
16435 ; Maximum stack depth per invocation: 18 words
16440 ;
16441 ;
16445 ;
16449 057500 T32::
16450 057500 004767 176252 1$: JSR PC,$T32 ;
16451 057504 104466 TRAP 66 ;
16452 057506 006000 ROR R0
16453 057510 103773 BLO 1$
16454 057512 000207 RTS PC
16455 ;
16456 ; Routine Size: 6 words
16457 ; Maximum stack depth per invocation: 0 words
16462 ;
16463 ;
16464 ; 7648 !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

7638
7642

7573

7450

7645

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (83)

```

16466 :ML4
16467 :
16468 :
16469 : 7649 !
16470 : 7650 !BGNTST:
16471 : 7651
16472 : 7652 !++
16473 : 7653 !TEST NUMBER: TST 33
16474 : 7654
16475 : 7655 !TEST NAME: SEQUENCER EXISTENCE TEST
16476 : 7656
16477 : 7657 !TEST DESCRIPTION:
16478 : 7658
16479 : 7659 !TEST SEQUENCER TIMING AND CONTROL
16480 : 7660 !LOGIC FOR EXISTENCE BY:
16481 : 7661
16482 : 7662 !1. FIRST WRITING THE GOOD BLOCK VIA DAT_DM MODE WITH A BACKGROUND
16483 : 7663 !PATTERN OF ONES.
16484 : 7664
16485 : 7665
16486 : 7666 !2. VIA A MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
16487 : 7667
16488 : 7668 !3. THEN VIA DAT_DM READ GOOD NIBBLES IN THE GOOD BLOCK AND XOR THEM
16489 : 7669 !AGAINST THE BACKGROUND PATTERN.
16490 : 7670
16491 : 7671 !RECORD THE NUMBER OF NIBBLES THAT WERE UNCHANGED OR PARTIALLY CHANGED
16492 : 7672 !BY THE MBUS WRITE.
16493 : 7673
16494 : 7674 !4. REPORT A FATAL ERROR AND DROP THIS UNIT IF THE NIBBLES TESTED EQUALS
16495 : 7675 !THE COUNT OF UNCHANGED NIBBLES.
16496 : 7676
16497 : 7677 !REPORT AN INTERMEDIATE DIAGNOSTIC MESSAGE IF AT LEAST SOME NIBBLES
16498 : 7678 !WERE CHANGED BY THE MBUS WRITE.
16499 : 7679
16500 : 7680 !IMPLICIT INPUTS:
16501 : 7681 !PD_TEMP:
16502 : 7682 !A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND ACCESSED FROM.
16503 : 7683
16504 : 7684 !IO_BUF:
16505 : 7685 !A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTIONS
16506 : 7686 !ARE FOUND.
16507 : 7687
16508 : 7688
16509 : 7689 local
16510 : 7690 !BG_PAT, !BACK GROUND PATTERN
16511 : 7691 !SUM_BAD, !SUM NIBBLE ARE BAD
16512 : 7692 !NIB_TSTED, !NIBBLES TESTED
16513 : 7693 !ALL_BAD, !ALL NIBBLES ARE BAD
16514 : 7694 !RESULT; !RESULTS OF XOR
16515 : 7695
16516 : 7696 CLR MBUS;
16517 : 7697 !BG_PAT = ONES; !BACKGROUND EQL ONES
16518 : 7698 !MLD1 = .BG_PAT; !LOAD DATA DIAG REG WITH BG PAT
16519 : 7699 !MLD2 = .BG_PAT;
16520 : 7700 !MLE2 = .BG_PAT;

```

```

16522 :ML4
16523 :
16524 :
16525 :       7701 DAT_DM_XFER ();
16526 :       7702 MLC51 = write;
16527 :       7703
16528 :       7704 incr WD_CNT from 0 to 127 do
16529 :       7705     begin
16530 :       7706     DELAY (ONE_US);
16531 :       7707     DAT_CLK = ONE;
16532 :       7708     end;
16533 :       7709
16534 :       7710 CLR_MBUS;
16535 :       7711 IO_BUF = ZEROES;
16536 :       7712 BAI = ONE;
16537 :       7713 GD_BLK_XFER ();
16538 :       7714 MLC51 = write;
16539 :       7715 TIME_OUT_LOOP;
16540 :       7716 BGNS_TB;
16541 :       7717 CLR_MBUS;
16542 :       7718 ALL_BAD = ZEROES;
16543 :       7719 SUM_BAD = ZEROES;
16544 :       7720 NIB_TSTED = ZEROES;
16545 :       7721 DAT_DM_XFER ();
16546 :       7722 MLC51 = read;
16547 :       7723 DELAY (ONE_US);
16548 :       7724
16549 :       7725 incr WD_CNT from 0 to 112 do
16550 :       7726     begin
16551 :       7727     PD_TEMP = .MLPD;
16552 :       7728     DAT_CLK = ONE;
16553 :       7729     DELAY (ONE_US);
16554 :       7730     RD_LNG_WRD;
16555 :       7731
16556 :       7732     incr NIB_PTR from 0 to 8 do
16557 :       7733
16558 :       7734     if .PD_TEMP [.NIB_PTR] eql ZERO
16559 :       7735     then
16560 :       7736     begin
16561 :       7737     NIB_TSTED = .NIB_TSTED + 1;
16562 :       7738     XOR_LNG_WRD (.NIB_PTR, .BG_PAT, RESULT);
16563 :       7739
16564 :       7740     if .RESULT<0, 4> eql ZERO
16565 :       7741     then
16566 :       7742     ALL_BAD = .ALL_BAD + 1
16567 :       7743     else
16568 :       7744
16569 :       7745     if .RESULT<0, 4> neq %0'17' then SUM_BAD = .SUM_BAD + 1;
16570 :       7746
16571 :       7747
16572 :       7748     end;
16573 :       7749
16574 :       7750     end;
16575 :       7751
16576 :       7752 ENDSUB;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (83)

!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

!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-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (83)

```

16578 :ML4
16579 :
16580 :
16581 : 7753
16582 : 7754 if .NIB_TSTED eql .ALL_BAD           !WHERE ALL NIBBLES XFERR'ED ALL BAD?
16583 : 7755 then                               !ERROR IF ALL BAD
16584 : 7756   begin
16585 : 7757   ERRDF (84, SYNC, DUMPER);
16586 : 7758   PRINTB (SEV_FMT, WRD_22, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
16587 : 7759   DODU (.ML_LDN);
16588 : 7760   DOCLN;
16589 : 7761   end
16590 : 7762 else
16591 : 7763
16592 : 7764   if .SUM_BAD gtr ZERO           !SEE IF SEE NIBBLE WERE BAD?
16593 : 7765   then
16594 : 7766   begin                           !SOME BAD IS OK SO GIVE INTERMEDIATE ERROR
16595 : 7767   ERRDF (85, INTER, DUMPER);
16596 : 7768   PRINTB (SIX_FMT, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
16597 : 7769   end;
16598 : 7770
16599 : 7771 ENDTST;

```

```

16607 057514 004167 124376          $T33: JSR      R1,SSAVES                ;
16608 057520 162706 000010          SUB      #10,SP                ;
16609 057524 152777 000040 133462  BISB    #40,@ML.REG+40        ;
16610 057532 016705 134044          MOV     ML,DUT,R5             ;
16611 057536 042705 177770          BIC    #177770,R5            ;
16612 057542 142777 000007 133444  BICB    #7,@ML.REG+40        ;
16613 057550 150577 133440          BISB    R5,@ML.REG+40        ;
16614 057554 012766 177777 000002  MOV     #-1,2(SP)             ; *,BG.PAT
16615 057562 012777 177777 133554  MOV     #-1,@ML.REG+170      ;
16616 057570 016677 000002 133556  MOV     2(SP),@ML.REG+200    ; BG.PAT,*
16617 057576 016677 000002 133530  MOV     2(SP),@ML.REG+160    ; BG.PAT,*
16618 057604 004767 134352          JSR    PC,DAI.DM.XFER        ;
16619 057610 012777 000061 133336  MOV     #61,@ML.REG         ;
16620 057616 005000          CLR    R0                    ; WD.CNT
16621 057620 012701 000001          1$:   MOV     #1,R1             ; *,SSTMP2
16622 057624 001411          2$:   BEQ     5$                ;
16623 057626 016702 122264          MOV     L$DLY,R2            ; *,SSTMP1
16624 057632 001404          BEQ     4$                    ;
16625 057634 005066 000006          3$:   CLR     6(SP)             ; SSTMP
16626 057640 005302          DEC     R2                    ; SSTMP1
16627 057642 001374          BNE    3$                    ;
16628 057644 005301          4$:   DEC     R1                ; SSTMP2
16629 057646 000766          BR     2$                    ;
16630 057650 152777 000020 133416  5$:   BISB    #20,@ML.REG+120    ;
16631 057656 005200          INC     R0                    ; WD.CNT

```


Address	OpCode	Operand1	Operand2	Comment	Label
16689					
16690					
16691					
16692	060152	006202		ASR R2	
16693	060154	062702	013120	ADD #PD.TEMP,R2	
16694	060160	010246		MOV R2,-(SP)	
16695	060162	010146		MOV R1,-(SP)	
16696	060164	042716	177770	BIC #177770,(SP)	: NIB.PTR,*
16697	060170	012746	000001	MOV #1,-(SP)	
16698	060174	005046		CLR -(SP)	
16699	060176	004767	122736	JSR PC,BLSGT2	
16700	060202	062706	000010	ADD #10,SP	
16701	060206	005700		TST R0	
16702	060210	001027		BNE 20\$	
16703	060212	005205		INC R5	: NIB.TSTED 7737
16704	060214	010146		MOV R1,-(SP)	: NIB.PTR,* 7738
16705	060216	016646	000004	MOV 4(SP),-(SP)	: BG.PAT,*
16706	060222	012746	000012	MOV #12,-(SP)	
16707	060226	060616		ADD SP,(SP)	: RESULT,*
16708	060230	004767	136406	JSR PC,XOR.LNG.WRD	
16709	060234	016602	000012	MOV 12(SP),R2	: RESULT,* 7740
16710	060240	042702	177760	BIC #177760,R2	
16711	060244	001002		BNE 18\$	
16712	060246	005204		INC R4	: ALL.BAD 7742
16713	060250	000405		BR 19\$: 7740
16714	060252	020227	000017	CMP R2,#17	: 7745
16715	060256	001402		BEQ 19\$	
16716	060260	005266	000006	INC 6(SP)	: SUM.BAD
16717	060264	062706	000006	ADD #6,SP	: 7736
16718	060270	005201		INC R1	: NIB.PTR 7732
16719	060272	020127	000010	CMP R1,#10	: NIB.PTR,*
16720	060276	003722		BLE 17\$	
16721	060300	005203		INC R3	: WD.CNT 7725
16722	060302	020327	000160	CMP R3,#160	: WD.CNT,*
16723	060306	003662		BLE 12\$	
16724	060310	104467		TRAP 67	: 7750
16725	060312	006000		ROR R0	
16726	060314	103615		BLO 7\$	
16727	060316	020504		CMP R5,R4	: NIB.TSTED,ALL.BAD 7754
16728	060320	001037		BNE 21\$	
16729	060322	104455		TRAP 55	: 7757
16730	060324	000124		.WORD 124	
16731	060326	010526		.WORD SYNC	
16732	060330	024052		.WORD DUMPER	
16733	060332	012746	006506	MOV #WRD.19,-(SP)	: 7758
16734	060336	012746	007774	MOV #FNC.5,-(SP)	
16735	060342	012746	006544	MOV #WRD.23,-(SP)	
16736	060346	012746	006426	MOV #WRD.12,-(SP)	
16737	060352	012746	006374	MOV #WRD.9,-(SP)	
16738	060356	012746	007454	MOV #PHR.4,-(SP)	
16739	060362	012746	006530	MOV #WRD.22,-(SP)	
16740	060366	012746	006120	MOV #SEV.FMT,-(SP)	
16741	060372	012746	000010	MOV #10,-(SP)	
16742	060376	010600		MOV SP,R0	: SP,*
16743	060400	104414		TRAP 14	

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

```

16745      ;ML4
16746      ;
16747
16748 060402 016700 133172      MOV      ML.LUN,R0      ;
16749 060406 104451      TRAP     51             ;
16750 060410 104444      TRAP     44             ;
16751 060412 062706 000022      ADD      #22,SP        ;
16752 060416 000432      BR       22$           ;
16753 060420 005716      21$:    TST      (SP)       ; SUM.BAD
16754 060422 003430      BLE     22$           ;
16755 060424 104455      TRAP     55             ;
16756 060426 000125      .WORD   125           ;
16757 060430 010672      .WORD   INTER         ;
16758 060432 024052      .WORD   DUMPER        ;
16759 060434 012746 006506      MOV      #WRD.19,-(SP) ;
16760 060440 012746 007774      MOV      #FNC.5,-(SP) ;
16761 060444 012746 006544      MOV      #WRD.23,-(SP) ;
16762 060450 012746 006426      MOV      #WRD.12,-(SP) ;
16763 060454 012746 006374      MOV      #WRD.9,-(SP)  ;
16764 060460 012746 007454      MOV      #PHR.4,-(SP) ;
16765 060464 012746 006100      MOV      #SIX.FMT,-(SP) ;
16766 060470 012746 000007      MOV      #7,-(SP)     ;
16767 060474 010600      MOV      SP,R0        ; SP,*
16768 060476 104414      TRAP     14           ;
16769 060500 062706 000020      ADD      #20,SP        ;
16770 060504 062706 000010      22$:    ADD      #10,SP   ;
16771 060510 000207      RTS      PC           ;
16772
16773      ; Routine Size: 255 words
16774      ; Maximum stack depth per invocation: 19 words
16779
16780
16784
16788 060512      T33::
16789 060512 004767 176776      1$:    JSR      PC,$T33 ;
16790 060516 104466      TRAP     66           ;
16791 060520 006000      ROR     R0            ;
16792 060522 103773      BLO     1$           ;
16793 060524 000207      RTS      PC           ;
16794
16795      ; Routine Size: 6 words
16796      ; Maximum stack depth per invocation: 0 words

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 374^{K 12}

SEQ 0361

16805
16806
16807 : 7772 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (84)

16809 :ML4
16810 :
16811 :
16812 :
16813 :
16814 :
16815 :
16816 :
16817 :
16818 :
16819 :
16820 :
16821 :
16822 :
16823 :
16824 :
16825 :
16826 :
16827 :
16828 :
16829 :
16830 :
16831 :
16832 :
16833 :
16834 :
16835 :
16836 :
16837 :
16838 :
16839 :
16840 :
16841 :
16842 :
16843 :
16844 :
16845 :
16846 :
16847 :
16848 :
16849 :
16850 :
16851 :
16852 :
16853 :
16854 :
16855 :
16856 :
16857 :
16858 :
16859 :
16860 :
16861 :
16862 :
16863 :

7773
7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785
7786
7787
7788
7789
7790
7791
7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824

! BGNTST;

! ++

! TEST NUMBER: TST 34

! 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 DA1 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 FROM DATA IS STORED
! AND ACCESSED FROM.

! IO_BUF

! A VECTOR OF 256 WORDS WHERE
! DATA FOR MBUS READS AND WRITE
! FUNCTION ARE FOUND.

! local

! DODU_FLG,
! BG_PAT,
! RESULT,
! TST_PAT;

! DROP UNIT FLAG
! BACKGROUND PATTERN
! RESULTS FROM XOR
! TEST PATTERN

```

16865 :ML4
16866 :
16867 :
16868 : 7825
16869 : 7826 DODU FLG = ZERO;
16870 : 7827 BG PAT = ZEROES;
16871 : 7828 TST_PAT = ONES;
16872 : 7829
16873 : 7830 incr TWICE from 0 to 1 do
16874 : 7831 begin
16875 : 7832   BGRSUB;
16876 : 7833   CLR_MBUS;
16877 : 7834   MLD1 = .BG_PAT;
16878 : 7835   MLD2 = .BG_PAT;
16879 : 7836   MLE2 = .BG_PAT;
16880 : 7837   DAT_DM_XFER ();
16881 : 7838   MLC51 = write;
16882 : 7839
16883 : 7840   incr WD_CNT from 0 to 127 do
16884 : 7841     begin
16885 : 7842       DELAY (ONE_US);
16886 : 7843       DAT_CLK = ONE;
16887 : 7844     end;
16888 : 7845
16889 : 7846   CLR_MBUS;
16890 : 7847   BAI = ONE;
16891 : 7848   IO_BUF = .TST_PAT;
16892 : 7849   GD_BLK_XFER ();
16893 : 7850   MLC51 = write;
16894 : 7851   TIME_OUT_LOOP;
16895 : 7852   CLR_MBUS;
16896 : 7853   DAT_DM_XFER ();
16897 : 7854   MLC51 = read;
16898 : 7855   DELAY (ONE_US);
16899 : 7856
16900 : 7857   incr WD_CNT from 0 to 112 do
16901 : 7858     begin
16902 : 7859       PD_TEMP = .MLPD;
16903 : 7860       DAT_CLK = ONE;
16904 : 7861       DELAY (ONE_US);
16905 : 7862       RD_LNG_WRD;
16906 : 7863
16907 : 7864       incr NIB_PTR from 0 to 8 do
16908 : 7865         !LOOK AT 9 NIBBLES
16909 : 7866         if .PD_TEMP [.NIB_PTR] IS_NOT_SET !FIND GOOD NIBBLES
16910 : 7867         then
16911 : 7868           begin
16912 : 7869             XOR_LNG_WRD (.NIB_PTR, .TST_PAT, RESULT); !XOR NIBBLE DATA WITH TST_PAT
16913 : 7870
16914 : 7871             if .RESULT<0, 4> neq ZERO !SEE IF EQUAL
16915 : 7872             then
16916 : 7873               begin !ERROR IF NEQ
16917 : 7874                 ERDF (86, SYNC, DUMPER);
16918 : 7875                 PRINTB (FIV_FMT, WRD 24, WRD 25, WRD 10, WRD 12, FNC_5);
16919 : 7876                 PRINTB (FMT_5, .TST_PAT, .RESULT, .NIB_PTR);

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (84)

```

!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
!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
!XOR NIBBLE DATA WITH TST_PAT
!SEE IF EQUAL
!ERROR IF NEQ

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (84)

```

16921 :ML4
16922 :
16923 :
16924 : 7877          DODU_FLG = ONE;
16925 : 7878          end
16926 : 7879
16927 : 7880          end
16928 : 7881
16929 : 7882          end;
16930 : 7883
16931 : 7884          ENDSUB;
16932 : 7885
16933 : 7886          if .DODU_FLG IS_SET
16934 : 7887
16935 : 7888          then
16936 : 7889              begin
16937 : 7890                  DODU (.ML_LUN);
16938 : 7891                  DOCLN;
16939 : 7892                  end;
16940 : 7893
16941 : 7894          BG PAT = not .BG PAT;
16942 : 7895          TST PAT = not .TST PAT;
16943 : 7896          end;
16944 : 7897
16945 : 7898          ENDTST;

```

!DROP THIS UNIT IF DODU IS_SET
!AND EXITS TEST

!COMPLIMENT BG PAT
!COMPLIMENT TST_PAT AND REPEAT

```

16953 060526 004167 123364      $T34: JSR    R1,$$SAVE5      :
16954 060532 024646             CMP    -(SP),-(SP)      :
16955 060534 005046             CLR    -(SP)           : DODU.FLG      7826
16956 060536 005001             CLR    R1              : BG.PAT       7827
16957 060540 012702 177777     MOV    #-1,R2          : *,TST.PAT   7828
16958 060544 005046             CLR    -(SP)           : TWICE       7830
16959 060546 104402             TRAP   2               :             7831
16960 060550 152777 000040 132436 $:    BISB  #40,@ML.REG+40  :             7832
16961 060556 016705 133020     MOV    ML.DUT,R5
16962 060562 042705 177770     BIC    #177770,R5
16963 060566 142777 000007 132420 $:    BICB  #7,@ML.REG+40
16964 060574 150577 132414     BISB  R5,@ML.REG+40
16965 060600 010177 132540     MOV    R1,@ML.REG+170  : BG.PAT,*    7834
16966 060604 010177 132544     MOV    R1,@ML.REG+200  : BG.PAT,*    7835
16967 060610 010177 132520     MOV    R1,@ML.REG+160  : BG.PAT,*    7836
16968 060614 004767 133342     JSR    PC,DAT.DM.XFER  :             7837
16969 060620 012777 000061 132326 $:    MOV    #61,@ML.REG
16970 060626 005003             CLR    R3              : WD.CNT      7838
16971 060630 012704 000001      2$:  MOV    #1,R4         : *,$$TMP2    7840
16972 060634 001411             3$:  BEQ    6$
16973 060636 016705 121254     MOV    LSDLY,R5        : *,$$TMP1    7842
16974 060642 001404             BEQ    5$

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

16976									22-Dec-1980 09:24:31	TOPS
16977									22-Dec-1980 09:21:22	PA:4
16978										
16979	060644	005066	000006		48:	CLR	6(SP)	:	SSTMP	
16980	060650	005305				DEC	R5	:	SSTMP1	
16981	060652	001374				BNE	48	:		
16982	060654	005304			58:	DEC	R4	:	SSTMP2	
16983	060656	000766				BR	38	:		
16984	060660	152777	000020	132406	68:	BISB	#20,2ML.REG+120	:		7843
16985	060666	005203				INC	R3	:	WD.CNT	7840
16986	060670	020327	000177			CMP	R3,#177	:	WD.CNT,*	
16987	060674	003755				BLE	28	:		
16988	060676	152777	000040	132310		BISB	#40,2ML.REG+40	:		7844
16989	060704	016705	132672			MOV	ML,DUT,R5	:		
16990	060710	042705	177770			BIC	#177770,R5	:		
16991	060714	142777	000007	132272		BICB	#7,2ML.REG+40	:		
16992	060722	150577	132266			BISB	R5,2ML.REG+40	:		
16993	060726	152777	000010	132260		BISB	#10,2ML.REG+40	:		7847
16994	060734	010267	130640			MOV	R2,IO.BUF	:	TST.PAT,*	7848
16995	060740	004767	135116			JSR	PC,GD.BLK.XFER	:		7849
16996	060744	012777	000061	132202		MOV	#61,2ML.REG	:		7850
16997	060752	105777	132246		78:	TSTB	2ML.REG+50	:		
16998	060756	100375				BPL	78	:		
16999	060760	152777	000040	132226		BISB	#40,2ML.REG+40	:		7851
17000	060766	016705	132610			MOV	ML,DUT,R5	:		
17001	060772	042705	177770			BIC	#177770,R5	:		
17002	060776	142777	000007	132210		BICB	#7,2ML.REG+40	:		
17003	061004	150577	132204			BISB	R5,2ML.REG+40	:		
17004	061010	004767	133146			JSR	PC,DAT.DM.XFER	:		7853
17005	061014	012777	000071	132132		MOV	#71,2ML.REG	:		7854
17006	061022	012704	000001			MOV	#1,R4	:	*,SSTMP2	7855
17007	061026	001411			88:	BEQ	118	:		
17008	061030	016705	121062			MOV	LSDLY,R5	:	*,SSTMP1	
17009	061034	001404				BEQ	108	:		
17010	061036	005066	000006		98:	CLR	6(SP)	:	SSTMP	
17011	061042	005305				DEC	R5	:	SSTMP1	
17012	061044	001374				BNE	98	:		
17013	061046	005304			108:	DEC	R4	:	SSTMP2	
17014	061050	000766				BR	88	:		
17015	061052	005003			118:	CLR	R3	:	WD.CNT	7857
17016	061054	017767	132324	132036	128:	MOV	2ML.REG+230,PD.TEMP	:		7859
17017	061062	152777	000020	132204		BISB	#20,2ML.REG+120	:		7860
17018	061070	012704	000001			MOV	#1,R4	:	*,SSTMP2	7861
17019	061074	001411			138:	BEQ	168	:		
17020	061076	016705	121014			MOV	LSDLY,R5	:	*,SSTMP1	
17021	061102	001404				BEQ	158	:		
17022	061104	005066	000006		148:	CLR	6(SP)	:	SSTMP	
17023	061110	005305				DEC	R5	:	SSTMP1	
17024	061112	001374				BNE	148	:		
17025	061114	005304			158:	DEC	R4	:	SSTMP2	
17026	061116	000766				BR	138	:		
17027	061120	017767	132220	130024	168:	MOV	2ML.REG+170,D1.TEMP	:		
17028	061126	017767	132222	130020		MOV	2ML.REG+200,D2.TEMP	:		
17029	061134	017767	132174	130014		MOV	2ML.REG+160,E2.TEMP	:		
17030	061142	005004				CLR	R4	:	NIB.PTR	7864

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

17032      :ML4
17033      :
17034      :
17035 061144 010405      17$: MOV R4,R5      ; NIB.PTR,*      7866
17036 061146 006205      ASR R5
17037 061150 006205      ASR R5
17038 061152 006205      ASR R5
17039 061154 062705 013120 ADD #PD.TEMP,R5
17040 061160 010546      MOV R5,-(SP)
17041 061162 010446      MOV R4,-(SP)      ; NIB.PTR,*
17042 061164 042716 177770 BIC #177770,(SP)
17043 061170 012746 000001 MOV #1,-(SP)
17044 061174 005046      CLR -(SP)
17045 061176 004767 121736 JSR PC,BLSGT2
17046 061202 062706 000010 ADD #10,SP
17047 061206 005700      TST R0
17048 061210 001060      BNE 19$
17049 061212 010446      MOV R4,-(SP)      ; NIB.PTR,*      7869
17050 061214 010246      MOV R2,-(SP)      ; TST.PAT,*
17051 061216 012746 000012 MOV #12,-(SP)
17052 061222 060616      ADD SP,(SP)      ; RESULT,*
17053 061224 004767 135412 JSR PC,XOR.LNG.WRD
17054 061230 032766 000017 000012 BIT #17,12(SP)      ; *,RESULT      7871
17055 061236 001443      BEQ 18$
17056 061240 104455      TRAP 55          ;          7874
17057 061242 000126      .WORD 126
17058 061244 010526      .WORD SYNC
17059 061246 024052      .WORD DUMPER
17060 061250 012746 007774 MOV #FNC.5,-(SP)
17061 061254 012746 006426 MOV #WRD.12,-(SP)
17062 061260 012746 006406 MOV #WRD.10,-(SP)
17063 061264 012746 006560 MOV #WRD.25,-(SP)
17064 061270 012746 006552 MOV #WRD.24,-(SP)
17065 061274 012746 006062 MOV #FIV.FMT,-(SP)
17066 061300 012746 000006 MOV #6,-(SP)
17067 061304 010600      MOV SP,R0      ; SP,*
17068 061306 104414      TRAP 14
17069 061310 010416      MOV R4,(SP)      ; NIB.PTR,*      7876
17070 061312 016646 000030 MOV 30(SP),-(SP)
17071 061316 010246      MOV R2,-(SP)      ; RESULT,*
17072 061320 012746 004430 MOV #FMT.5,-(SP)      ; TST.PAT,*
17073 061324 012746 000004 MOV #4,-(SP)
17074 061330 010600      MOV SP,R0      ; SP,*
17075 061332 104414      TRAP 14
17076 061334 012766 000001 000036 MOV #1,36(SP)
17077 061342 062706 000026 ADD #26,SP      ; *,DODU.FLG      7877
17078 061346 062706 000006 ADD #6,SP          ;          7873
17079 061352 005204      18$: INC R4          ;          7868
17080 061354 020427 000010 19$: CMP R4,#10      ; NIB.PTR      7864
17081 061360 003671      BLE 17$          ; NIB.PTR,*
17082 061362 005203      INC R3          ; WD.CNT      7857
17083 061364 020327 000160 CMP R3,#160      ; WD.CNT,*
17084 061370 003631      BLE 12$
17085 061372 104467      TRAP 67          ;          7882
17086 061374 006000      ROR R0
    
```

```

17088 ;ML4
17089 ;
17090
17091 061376 103002
17092 061400 000167 177142
17093 061404 026627 000002 000001 20$: BHIS 21$
17094 061412 001004 21$: JMP 1$
17095 061414 016700 132160 BNE 22$ ; DODU.FLG,* 7886
17096 061420 104451 MOV ML.LUN,R0 ; 7890
17097 061422 104444 TRAP 51
17098 061424 005101 22$: COM R1 ; BG.PAT 7894
17099 061426 005102 COM R2 ; TST.PAT 7895
17100 061430 005216 INC (SP) ; TWICE 7830
17101 061432 021627 000001 CMP (SP),#1 ; TWICE,*
17102 061436 003760 BLE 20$
17103 061440 062706 000010 ADD #10,SP ;
17104 061444 000207 RTS PC ; 7771
17105
17106 ; Routine Size: 232 words
17107 ; Maximum stack depth per invocation: 24 words
17112
17113
17117
17121 061446 T34::
17122 061446 004767 177054 1$: JSR PC,$T34 ; 7896
17123 061452 104466 TRAP 66
17124 061454 006000 ROR R0
17125 061456 103773 BLO 1$
17126 061460 000207 RTS PC
17127
17128 ; Routine Size: 6 words
17129 ; Maximum stack depth per invocation: 0 words
17134
17135
17136 ; 7899 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (85)

```

17138 :ML4
17139 :
17140 :
17141 : 7900 !
17142 : 7901 !BGNTST;
17143 : 7902 !
17144 : 7903 !++
17145 : 7904 !TEST NUMBER: TST 35
17146 : 7905 !
17147 : 7906 !TEST NAME: SYNC DATA BUS CONTINUITY /READ PATH
17148 : 7907 !
17149 : 7908 !TEST DESCRIPTION:
17150 : 7909 !TEST THE CONTINUITY OF THE SYNCHRONOUS MODULE READ
17151 : 7910 !DATA BUS BY:
17152 : 7911 !
17153 : 7912 ! 1. VIA MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
17154 : 7913 !
17155 : 7914 ! 2. VIA MBUS READ FUNCTION READ THE GOOD BLOCK FOR ONES.
17156 : 7915 !
17157 : 7916 ! 3. REPEAT WITH COMPLIMENT DATA PATTERN.
17158 : 7917 !
17159 : 7918 !IMPLICIT INPUTS:
17160 : 7919 !IO_BUF
17161 : 7920 !A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
17162 : 7921 !FUNCTION ARE FOUND.
17163 : 7922 !
17164 : 7923 !
17165 : 7924 !local
17166 : 7925 !DODU_FLG, !DROP UNIT FLAG
17167 : 7926 !TST_PAT, !TEST PATTERN
17168 : 7927 !BG_PAT; !BACKGROUND PATTERN
17169 : 7928 !
17170 : 7929 !DODU_FLG = ZERO;
17171 : 7930 !TST_PAT = ONES;
17172 : 7931 !BG_PAT = ZEROES;
17173 : 7932 !
17174 : 7933 !incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
17175 : 7934 !begin
17176 : 7935 !CLR_MBUS;
17177 : 7936 !BAI = ONE; !SET ON FIRST IO_BUF ADRS
17178 : 7937 !IO_BUF = .TST_PAT; !FIRST IO_BUF ADRS GET TST_PAT
17179 : 7938 !GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
17180 : 7939 !MLCS1 = write; !DO A WRITE FUNCTION (WRITES THE TST_PAT)
17181 : 7940 !TIME_OUT_LOOP;
17182 : 7941 !BGNSDB;
17183 : 7942 !CLR_MBUS;
17184 : 7943 !
17185 : 7944 !incr IO_CNT from 0 to 255 do !LOAD IO_BUF WITH BG PAT
17186 : 7945 !IO_BUF [.IO_CNT] = .BG_PAT;
17187 : 7946 !
17188 : 7947 !GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
17189 : 7948 !MLCS1 = read; !DO A READ FUNCTION (READ THE TST_PAT)
17190 : 7949 !TIME_OUT_LOOP;
17191 : 7950 !
17192 : 7951 !incr IO_CNT from 0 to 255 do !READ THE IO_BUF FOR TEST PATTERN

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (85)

```

17194 :ML4
17195 :
17196 :
17197 : 7952
17198 : 7953
17199 : 7954
17200 : 7955
17201 : 7956
17202 : 7957
17203 : 7958
17204 : 7959
17205 : 7960
17206 : 7961
17207 : 7962
17208 : 7963
17209 : 7964
17210 : 7965
17211 : 7966
17212 : 7967
17213 : 7968
17214 : 7969
17215 : 7970
17216 : 7971
17217 : 7972
17218 : 7973
17219 : 7974
17220 : 7975
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 :
17246 :
17247 :

```

```

if .IO_BUF [.IO_CNT] neq .TST_PAT      !SEE IF IO_BUF WORD EQLS TST PAT
then
begin                                  !ERROR AND SET DODU_FLG IF NEQ
ERRDF (87, SYNC, DUMPER);
PRINTB (SEV_FMT, WRD 24, WRD 25, WRD 10, WRD 12, WRD 23, FNC 6, WRD 19);
PRINTB (FMT_2, .TST_PAT, .IO_BUF [.IO_CNT], (.TST_PAT xor .IO_BUF [.IO_CNT]));
DODU_FLG = ONE;
end;
ENDSUB;

if .DODU_FLG IS_SET                    !DROP THIS UNIT IF DODU_FLG IS SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;

TST_PAT = not .TST_PAT;                !COMPLIMENT TST_PAT
BG_PAT = not .BG_PAT;                  !COMPLIMENT BG_PAT AND REPEAT
end;
ENDTST;

```

```

17228 061462 004167 122430      $T35: JSR      R1,$SAVE5          :
17229 061466 005046 177777      CLR      -(SP)          : DODU.FLG
17230 061470 012701 177777      MOV      #-1,R1        : *,TST.PAT
17231 061474 005004 177777      CLR      R4            : BG.PAT
17232 061476 005005 177777      CLR      R5            : TWICE
17233 061500 152777 000040 131506 1$: BISB     #40,2ML.REG+40 :
17234 061506 016703 132070 177777      MOV      ML.DUT,R3     :
17235 061512 042703 177770 177777      BIC      #177770,R3    :
17236 061516 142777 000007 131470 177777      BICB     #7,2ML.REG+40 :
17237 061524 150377 131464 177777      BISB     R3,2ML.REG+40 :
17238 061530 152777 000010 131456 177777      BISB     #10,2ML.REG+40 :
17239 061536 010167 130036 177777      MOV      R1,IO.BUF     : TST.PAT,*
17240 061542 004767 134314 177777      JSR      PC,GD.BLK.XFER :
17241 061546 012777 000061 131400 177777      MOV      #61,2ML.REG  :
17242 061554 105777 131444 177777      TSTB     2ML.REG+50   :
17243 061560 100375 177777      BPL      2$           :
17244 061562 104402 177777      TRAP     2            :
17245 061564 152777 000040 131422 177777      BISB     #40,2ML.REG+40 :
17246 061572 016703 132004 177777      MOV      ML.DUT,R3     :
17247 061576 042703 177770 177777      BIC      #177770,R3    :

```

7898
7929
7930
7931
7933
7934

7936
7937
7938
7939

7940
7941

Address	OpCode	Operand 1	Operand 2	Operand 3	Comment	IO.CNT	IO.CNT*	Other
17249					:ML4			
17250					:			
17251								
17252	061602	142777	000007	131404	BICB #7,@ML.REG+40			
17253	061610	150377	131400		BISB R3,@ML.REG+40			
17254	061614	005002			CLR R2	: IO.CNT		7944
17255	061616	010203			4\$: MOV R2,R3	: IO.CNT,*		7945
17256	061620	006303			ASL R3			
17257	061622	010463	011600		MOV R4,IO.BUF(R3)	: BG.PAT,*		
17258	061626	005202			INC R2	: IO.CNT		7944
17259	061630	020227	000377		CMP R2,#377	: IO.CNT,*		
17260	061634	003770			BLE 4\$			
17261	061636	004767	134220		JSR PC,GD.BLK.XFER	:		7947
17262	061642	012777	000071	131304	MOV #71,@ML.REG	:		7948
17263	061650	105777	131350		5\$: TSTB @ML.REG+50			
17264	061654	100375			BPL 5\$			
17265	061656	005002			CLR R2	: IO.CNT		7951
17266	061660	010203			6\$: MOV R2,R3	: IO.CNT,*		7953
17267	061662	006303			ASL R3			
17268	061664	062703	011600		ADD #IO.BUF,R3			
17269	061670	021301			CMP (R3),R1	: *,TST.PAT		
17270	061672	001454			BEQ 7\$			
17271	061674	104455			TRAP 55	:		7956
17272	061676	000127			.WORD 127			
17273	061700	010526			.WORD SYNC			
17274	061702	024052			.WORD DUMPER			
17275	061704	012746	006506		MOV #WRD.19,-(SP)	:		7957
17276	061710	012746	010004		MOV #FNC.6,-(SP)			
17277	061714	012746	006544		MOV #WRD.23,-(SP)			
17278	061720	012746	006426		MOV #WRD.12,-(SP)			
17279	061724	012746	006406		MOV #WRD.10,-(SP)			
17280	061730	012746	006560		MOV #WRD.25,-(SP)			
17281	061734	012746	006552		MOV #WRD.24,-(SP)			
17282	061740	012746	006120		MOV #SEV.FMT,-(SP)			
17283	061744	012746	000010		MOV #10,-(SP)			
17284	061750	010600			MOV SP,R0	: SP,*		
17285	061752	104414			TRAP 14			
17286	061754	011316			MOV (R3),(SP)	:		7958
17287	061756	010146			MOV R1,-(SP)	: TST.PAT,*		
17288	061760	046616	000002		BIC 2(SP),(SP)	: TST.PAT,*		
17289	061764	040166	000002		BIC R1,2(SP)			
17290	061770	052616			BIS (SP)+,(SP)			
17291	061772	011346			MOV (R3)-,(SP)			
17292	061774	010146			MOV R1,-(SP)	: TST.PAT,*		
17293	061776	012746	004266		MOV #FMT.2,-(SP)			
17294	062002	012746	000004		MOV #4,-(SP)			
17295	062006	010600			MOV SP,R0	: SP,*		
17296	062010	104414			TRAP 14			
17297	062012	012766	000001	000032	MOV #1,32(SP)	: *,DODU.FLG		7959
17298	062020	062706	000032		ADD #32,SP	:		7955
17299	062024	005202			7\$: INC R2	: IO.CNT		7951
17300	062026	020227	000377		CMP R2,#377	: IO.CNT,*		
17301	062032	003712			BLE 6\$			
17302	062034	104467			TRAP 67	:		7960
17303	062036	006000			ROR R0			

```
17305      ;ML4
17306      ;
17307
17308 062040 103650      BLO      3$
17309 062042 021627 000001  CMP      (SP),#1      ; DODU.FLG,*      7964
17310 062046 001004      BNE      8$
17311 062050 016700 131524  MOV      ML,LUN,R0      ;
17312 062054 104451      TRAP     51
17313 062056 104444      TRAP     44
17314 062060 005101      8$:      COM      R1      ; TST.PAT      7971
17315 062062 005104      COM      R4      ; BG.PAT      7972
17316 062064 005205      INC      R5      ; TWICE      7933
17317 062066 020527 000001  CMP      R5,#1      ; TWICE,*
17318 062072 003602      BLE     1$
17319 062074 005726      TST     (SP)+
17320 062076 000207      RTS     PC      ;
17321
17322      ; Routine Size: 135 words
17323      ; Maximum stack depth per invocation: 20 words
17328
17329
17333
17337 062100
17338 062100 004767 177356  T35::
1$:      JSR     PC,$T35      ;
17339 062104 104466      TRAP     66
17340 062106 006000      ROR     R0
17341 062110 103773      BLO     1$
17342 062112 000207      RTS     PC
17343
17344      ; Routine Size: 6 words
17345      ; Maximum stack depth per invocation: 0 words
17350
17351
17352 :      7976 !<BLF/PAGE>
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (86)

17354 :ML4
17355 :
17356 :
17357 : 7977
17358 : 7978
17359 : 7979
17360 : 7980
17361 : 7981
17362 : 7982
17363 : 7983
17364 : 7984
17365 : 7985
17366 : 7986
17367 : 7987
17368 : 7988
17369 : 7989
17370 : 7990
17371 : 7991
17372 : 7992
17373 : 7993
17374 : 7994
17375 : 7995
17376 : 7996
17377 : 7997
17378 : 7998
17379 : 7999
17380 : 8000
17381 : 8001
17382 : 8002
17383 : 8003
17384 : 8004
17385 : 8005
17386 : 8006
17387 : 8007
17388 : 8008
17389 : 8009
17390 : 8010
17391 : 8011
17392 : 8012
17393 : 8013
17394 : 8014
17395 : 8015
17396 : 8016
17397 : 8017
17398 : 8018
17399 : 8019
17400 : 8020
17401 : 8021
17402 : 8022
17403 : 8023
17404 : 8024
17405 : 8025
17406 : 8026
17407 : 8027
17408 : 8028

BGNTST;

!++

TEST NUMBER: TST 36

TEST NAME: RAM BUS ADRS COUNTER TEST /WRITE PATH

TEST DESCRIPTION:

TEST ABILITY OF THE RAM_BUS ADDRESS
COUNTERS TO LOAD/UNLOAD THE SKIP
RAM DURING WRITE 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. VIA DAT DM READ GOOD NIBBLES IN
THE GOOD BLOCK FOR THE UNBROKEN
COUNT OF 0 TO 63.

ONCE A BAD NIBBLE IS ENCOUNTERED
MASK THAT NIBBLE FROM FURTHER
READS.
4. REPEAT READING NIBBLES UNTIL
113 WORDS ARE READ OR ALL 10 NIBBLES
ARE MASKED.

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
FUNCTION ARE FOUND.

local

DODU_FLG,

!DROP UNIT FLAG

```

17410 :ML4
17411 :
17412 :
17413 :      8029      WRD_CNT,           !WORD COUNT
17414 :      8030      NIB_IGNORE : bitvector [16], !NIBBLE IGNORE FLAGS
17415 :      8031      ERR_FLG,           !ERROR FLAG
17416 :      8032      BAD_NIB_CNT,       !BAD NIBBLE COUNT
17417 :      8033      PASS_CNT,         !PASS COUNT
17418 :      8034      NIB_PAT;          !NIBBLE PATTERN
17419 :      8035
17420 :      8036      DODU_FLG = ZERO;
17421 :      8037      WRD_CNT = ZERO;
17422 :      8038
17423 :      8039      incr CNT from 0 to 63 do           !LOAD 64 WORDS WITH REPEATING COUNTS OF 0-63
17424 :      8040
17425 :      8041      incr PAT_CNT from -1 to 11 by 4 do !LOAD NIBBLES IN WORD WITH REPEATING COUNTS OF 0-63
17426 :      8042      begin
17427 :      8043      (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1; !LOAD FIRST NIBBLE IN WORD
17428 :      8044      (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2; !LOAD SECOND NIBBLE IN WORD
17429 :      8045      (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3; !LOAD THIRD NIBBLE IN WORD
17430 :      8046      (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4; !LOAD FORTH NIBBLE IN WORD
17431 :      8047      WRD_CNT = .WRD_CNT + 1;           !INCREMENT TO NEXT WORD
17432 :      8048      end;
17433 :      8049
17434 :      8050      BGNSUB;
17435 :      8051      CLR MBUS;
17436 :      8052      GD_BLK_XFER ();                 !SET UP A GOOD BLOCK XFERR
17437 :      8053      MLCS1 = write;                  !DO A WRITE FUNCTION
17438 :      8054      TIME_OUT_LOOP;
17439 :      8055      CLR MBUS;
17440 :      8056      NIB_IGNORE = ZEROES;
17441 :      8057      PASS_CNT = ZEROES;
17442 :      8058      NIB_PAT = ZEROES;
17443 :      8059      BAD_NIB_CNT = ZEROES;
17444 :      8060      DAT_DM_XFER ();                 !SET UP A DATA DIAG MODE AT THE GOOD BLOCK
17445 :      8061      MLCS1 = read;                   !DO A READ FUNCTION
17446 :      8062      DELAY (ONE_US);
17447 :      8063
17448 :      8064      do
17449 :      8065      begin
17450 :      8066      PD_TEMP = .MLPD;
17451 :      8067      DAT_CLK = ONE;
17452 :      8068      DELAY (ONE_US);
17453 :      8069      RD_LNG_WRD;
17454 :      8070
17455 :      8071      incr NIB_PTR from 0 to 8 do       !LOOP UNTIL THE BLOCK IS READ OR 9 BAD NIBBLES FOUND
17456 :      8072      begin
17457 :      8073
17458 :      8074      if .PD_TEMP [.NIB_PTR] IS_NOT_SET !FIND GOOD NIBBLES
17459 :      8075      then
17460 :      8076      begin
17461 :      8077
17462 :      8078      if .NIB_IGNORE [.NIB_PTR] IS_NOT_SET !SEE IF THIS NIBBLE FOUND BAD BEFORE
17463 :      8079      then
17464 :      8080      begin

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (86)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (86)

```

17466 :ML4
17467 :
17468 :
17469 :      8081      TST_LNG_WRD (.NIB_PTR, .NIB_PAT, ERR_FLG);      !TEST THE NIBBLE IF NEVER FOUND BAD
17470 :      8082
17471 :      8083      if .ERR_FLG IS_SET      !SEE IF TEST FOUND AN ERROR
17472 :      8084      then
17473 :      8085      begin      !ERROR AND SET DODU_FLG IS SET
17474 :      8086      ERRDF (88, ARR_DAT, DUMPER);
17475 :      8087      PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_5, WRD_19);
17476 :      8088      DODU_FLG = ONE;
17477 :      8089      end
17478 :      8090
17479 :      8091      end
17480 :      8092
17481 :      8093      end
17482 :      8094      else      !THIS NIBBLE IS BAD
17483 :      8095      begin
17484 :      8096      NIB_IGNORE [.NIB_PTR] = ONE;      !SET THIS NIBBLE NIB_IGNORE FLAG
17485 :      8097      BAD_NIB_CNT = .BAD_NIB_CNT + 1;      !INCREMENT BAD NIB COUNT
17486 :      8098      end;
17487 :      8099
17488 :      8100      NIB_PAT = .NIB_PAT + 1;      !INCREMENT NIB PAT
17489 :      8101      end;
17490 :      8102
17491 :      8103      PASS_CNT = .PASS_CNT + 1;      !INCREMENT PASS COUNT
17492 :      8104      end
17493 :      8105      until (.PASS_CNT eql 113) or (.BAD_NIB_CNT eql 9);      !REPEAT UNTIL COMPLETE
17494 :      8106
17495 :      8107      ENDSUB;
17496 :      8108
17497 :      8109      if .DODU_FLG IS_SET      !DROP THIS UNIT IF DODU_FLG SET
17498 :      8110      then
17499 :      8111      begin
17500 :      8112      DODU (.ML_LUN);
17501 :      8113      DOCLN;
17502 :      8114      end;
17503 :      8115
17504 :      8116      ENDTST;
17508 :
17512 062114 004167 121776      $T36:      JSR      R1, $SAVE5      ;
17513 062120 162706 000006      SUB      #6, SP      ;
17514 062124 005046      CLR      -(SP)      ; DODU.FLG      8036
17515 062126 005003      CLR      R3      ; WRD.CNT      8037
17516 062130 005002      CLR      R2      ; CNT      8039
17517 062132 012700 177777      1$:      MOV      #-1, R0      ; *,PAT.CNT      8041
17518 062136 010301      2$:      MOV      R3, R1      ; WRD.CNT, *      8043
17519 062140 006301      ASL      R1

```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

17521										
17522										
17523										
17524	062142	062701	011600							
17525	062146	010005								
17526	062150	005205								
17527	062152	042705	177760							
17528	062156	142711	000017							
17529	062162	150511								
17530	062164	010005								
17531	062166	062705	000002							8044
17532	062172	006305								
17533	062174	006305								
17534	062176	006305								
17535	062200	006305								
17536	062202	042705	177417							
17537	062206	142711	000360							
17538	062212	150511								
17539	062214	010005								
17540	062216	062705	000003							8045
17541	062222	000305								
17542	062224	042705	170377							
17543	062230	042711	007400							
17544	062234	050511								
17545	062236	010005								
17546	062240	062705	000004							8046
17547	062244	000305								
17548	062246	006305								
17549	062250	006305								
17550	062252	006305								
17551	062254	006305								
17552	062256	042705	007777							
17553	062262	042711	170000							
17554	062266	050511								
17555	062270	005203								
17556	062272	062700	000004							8047
17557	062276	020027	000013							8041
17558	062302	003715								
17559	062304	005202								
17560	062306	020227	000077							8039
17561	062312	003707								
17562	062314	104402								
17563	062316	152777	000040	130670	3\$:					8048
17564	062324	016702	131252							8050
17565	062330	042702	177770							
17566	062334	142777	000007	130652						
17567	062342	150277	130646							
17568	062346	004767	133510							
17569	062352	012777	000061	130574						8052
17570	062360	105777	130640		4\$:					8053
17571	062364	100375								
17572	062366	152777	000040	130620						8054
17573	062374	016702	131202							
17574	062400	042702	177770							
17575	062404	142777	000007	130602						

```

17577      ;ML4
17578      ;
17579
17580 062412 150277 130576      BISB   R2,@ML.REG+40
17581 062416 005066 000004      CLR    4(SP)
17582 062422 005004                CLR    R4
17583 062424 005003                CLR    R3
17584 062426 005005                CLR    R5
17585 062430 004767 131526      JSR    PC,DAT.DM.XFER
17586 062434 012777 000071 130512  MOV    #71,@ML.REG
17587 062442 012701 000001      MOV    #1,R1
17588 062446 001411      5$:   BEQ    8$
17589 062450 016702 117442      MOV    L$DLY,R2
17590 062454 001404                BEQ    7$
17591 062456 005066 000006      6$:   CLR    6(SP)
17592 062462 005302                DEC    R2
17593 062464 001374                BNE    6$
17594 062466 005301      7$:   DEC    R1
17595 062470 000766                BR     5$
17596 062472 017767 130706 130420 8$:   MOV    @ML.REG+230,PD.TEMP
17597 062500 152777 000020 130566  BISB   #20,@ML.REG+120
17598 062506 012701 000001      MOV    #1,R1
17599 062512 001411      9$:   BEQ    12$
17600 062514 016702 117376      MOV    L$DLY,R2
17601 062520 001404                BEQ    11$
17602 062522 005066 000006      10$:  CLR    6(SP)
17603 062526 005302                DEC    R2
17604 062530 001374                BNE    10$
17605 062532 005301      11$:  DEC    R1
17606 062534 000766                BR     9$
17607 062536 017767 130602 126406 12$:  MOV    @ML.REG+170,D1.TEMP
17608 062544 017767 130604 126402  MOV    @ML.REG+200,D2.TEMP
17609 062552 017767 130556 126376  MOV    @ML.REG+160,E2.TEMP
17610 062560 005001                CLR    R1
17611 062562 010100      13$:  MOV    R1,R0
17612 062564 006200                ASR    R0
17613 062566 006200                ASR    R0
17614 062570 006200                ASR    R0
17615 062572 012702 000004      MOV    #4,R2
17616 062576 060602                ADD    SP,R2
17617 062600 060002                ADD    R0,R2
17618 062602 010046                MOV    R0,-(SP)
17619 062604 062716 013120      ADD    #PD.TEMP,(SP)
17620 062610 010146                MOV    R1,-(SP)
17621 062612 042716 177770      BIC    #177770,(SP)
17622 062616 012746 000001      MOV    #1,-(SP)
17623 062622 005046                CLR    -(SP)
17624 062624 004767 120310      JSR    PC,BLSGT2
17625 062630 062706 000010      ADD    #10,SP
17626 062634 005700                TST    R0
17627 062636 001066                BNE    15$
17628 062640 010246                MOV    R2,-(SP)
17629 062642 010146                MOV    R1,-(SP)
17630 062644 042716 177770      BIC    #177770,(SP)
17631 062650 012746 000001      MOV    #1,-(SP)

```

```

: NIB.IGNORE
: PASS.CNT
: NIB.PAT
: BAD.NIB.CNT
:
: *,SSTMP2
: *,SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: *,SSTMP2
: *,SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: *,SSTMP2
: *,SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
:
: NIB.PTR
: NIB.PTR,*
:
: NIB.IGNORE,*
:
: NIB.PTR,*
:
: NIB.PTR,*

```

```

8056
8057
8058
8059
8060
8061
8062
8066
8067
8068
8071
8074
8078
8074
8078

```

Address	OpCode	Operand 1	Operand 2	Operand 3	Label	Instruction	Comments	Line No.
17633					:ML4			
17634					:			
17635								
17636	062654	005046				CLR	-(SP)	
17637	062656	004767	120256			JSR	PC,BLSGT2	
17638	062662	062706	000010			ADD	#10,SP	
17639	062666	005700				TST	R0	
17640	062670	001065				BNE	16\$	
17641	062672	010146				MOV	R1,-(SP)	: NIB.PTR,*
17642	062674	010346				MOV	R3,-(SP)	: NIB.PAT,*
17643	062676	012746	000010			MOV	#10,-(SP)	
17644	062702	060616				ADD	SP,(SP)	: ERR.FLG,*
17645	062704	004767	133236			JSR	PC,TST.LNG.WRD	
17646	062710	026627	000010	000001		CMP	10(SP),#1	: ERR.FLG,*
17647	062716	001033				BNE	14\$	
17648	062720	104455				TRAP	55	:
17649	062722	000130				.WORD	130	
17650	062724	010570				.WORD	ARR.DAT	
17651	062726	024052				.WORD	DUMPER	
17652	062730	012746	006506			MOV	#WRD.19,-(SP)	:
17653	062734	012746	007774			MOV	#FNC.5,-(SP)	
17654	062740	012746	006426			MOV	#WRD.12,-(SP)	
17655	062744	012746	006406			MOV	#WRD.10,-(SP)	
17656	062750	012746	007046			MOV	#WRD.50,-(SP)	
17657	062754	012746	010172			MOV	#FNC.18,-(SP)	
17658	062760	012746	006100			MOV	#SIX.FMT,-(SP)	
17659	062764	012746	000007			MOV	#7,-(SP)	
17660	062770	010600				MOV	SP,R0	: SP,*
17661	062772	104414				TRAP	14	
17662	062774	012766	000001	000026		MOV	#1,26(SP)	: *,DODU.FLG
17663	063002	062706	000020			ADD	#20,SP	:
17664	063006	062706	000006		14\$:	ADD	#6,SP	:
17665	063012	000414				BR	16\$:
17666	063014	010246			15\$:	MOV	R2,-(SP)	:
17667	063016	010146				MOV	R1,-(SP)	: NIB.PTR,*
17668	063020	042716	177770			BIC	#177770,(SP)	
17669	063024	012746	000001			MOV	#1,-(SP)	
17670	063030	011646				MOV	(SP),-(SP)	
17671	063032	004767	120340			JSR	PC,BLSPU2	
17672	063036	005205				INC	R5	: BAD.NIB.CNT
17673	063040	062706	000010			ADD	#10,SP	:
17674	063044	005203			16\$:	INC	R3	: NIB.PAT
17675	063046	005201				INC	R1	: NIB.PTR
17676	063050	020127	000010			CMP	R1,#10	: NIB.PTR,*
17677	063054	003642				BLE	13\$	
17678	063056	005204				INC	R4	: PASS.CNT
17679	063060	020427	000161			CMP	R4,#161	: PASS.CNT,*
17680	063064	001405				BEQ	17\$	
17681	063066	020527	000011			CMP	R5,#11	
17682	063072	001402				BEQ	17\$	
17683	063074	000167	177372			JMP	8\$	
17684	063100	104467			17\$:	TRAP	67	
17685	063102	006000				ROR	R0	
17686	063104	103002				BHIS	18\$	
17687	063106	000167	177202			JMP	3\$	

1
6
2
4
5
6
7
8
0
2

```
17689 ;ML4
17690 ;
17691 ;
17692 063112 021627 000001 18$: CMP (SP),#1 ; DODU.FLG,* 8109
17693 063116 001004 BNE 19$
17694 063120 016700 130454 MOV ML,LUN,R0 ; 8112
17695 063124 104451 TRAP 51
17696 063126 104444 TRAP 44
17697 063130 062706 000010 19$: ADD #10,SP ; 7975
17698 063134 000207 RTS PC
17699
17700 ; Routine Size: 265 words
17701 ; Maximum stack depth per invocation: 21 words
17706
17707
17711
17715 063136 T36::
17716 063136 004767 176752 1$: JSR PC,$T36 ; 8114
17717 063142 104466 TRAP 66
17718 063144 006000 ROR R0
17719 063146 103773 BLO 1$
17720 063150 000207 RTS PC
17721
17722 ; Routine Size: 6 words
17723 ; Maximum stack depth per invocation: 0 words
17728
17729
17730 : 8117 !<BLF/PAGE>
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (87)

17732 :ML4
17733 :
17734 :
17735 :
17736 :
17737 :
17738 :
17739 :
17740 :
17741 :
17742 :
17743 :
17744 :
17745 :
17746 :
17747 :
17748 :
17749 :
17750 :
17751 :
17752 :
17753 :
17754 :
17755 :
17756 :
17757 :
17758 :
17759 :
17760 :
17761 :
17762 :
17763 :
17764 :
17765 :
17766 :
17767 :
17768 :
17769 :
17770 :
17771 :
17772 :
17773 :
17774 :
17775 :
17776 :
17777 :
17778 :
17779 :
17780 :
17781 :
17782 :
17783 :
17784 :
17785 :
17786 :

8118
8119
8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8150
8151
8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169

```

:
:
: BGNTST;
:
: ++
: TEST NUMBER: TST 37
:
: 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.
:
:
: local
: DODU_FLG,
: WRD_CNT,
: PAT_INC,
: SIZ_EXP,
: POS_EXP,
: TEMP;
:
: CLR MBUS;
: DODU_FLG = ZERO;
: WRD_CNT = ZERO;
:
: incr COUNT from 0 to 63 do

```

```

!DROP UNIT FLAG
!WORD COUNT
!PATTERN INCREMENT
!SIZE EXPRESSION
!POSITIONAL EXPRESSION
!TEMPORARY STORAGE LOCATION

```

!LOAD 64 WORDS WITH REPEATING COUNTS OF 0-63

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 FA:<NEALE>BL3ML4.BLI.2 (87)

```

17788 :ML4
17789 :
17790 :
17791 :      8170
17792 :      8171      incr PAT_CNT from -1 to 11 by 4 do      !LOAD NIBBLES IN WORD WITH REPEATING COUNT OF 0-63
17793 :      8172      begin
17794 :      8173      (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1;      !LOAD FIRST NIBBLE IN WORD
17795 :      8174      (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2;      !LOAD SECOND NIBBLE IN WORD
17796 :      8175      (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3;      !LOAD THIRD NIBBLE IN WORD
17797 :      8176      (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4;      !LOAD FORTH NIBBLE IN WORD
17798 :      8177      WRD_CNT = .WRD_CNT + 1;      !INCREMENT TO NEXT WORD
17799 :      8178      end;
17800 :      8179
17801 :      8180      GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
17802 :      8181      MLCS1 = write;      !DO A WRITE FUNCTION
17803 :      8182      TIME_OUT_LOOP;
17804 :      8183      BGNSOB;
17805 :      8184
17806 :      8185      incr IO_CNT from 0 to 255 do      !CLEAR OUT IO_BUF
17807 :      8186      IO_BUF [.IO_CNT] = ZEROES;
17808 :      8187
17809 :      8188      CLR_MBUS;
17810 :      8189      GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
17811 :      8190      MLCS1 = read;      !DO A READ FUNCTION
17812 :      8191      TIME_OUT_LOOP;
17813 :      8192      CLR_MBUS;
17814 :      8193      SIZ_EXP = 4;      !FIELD SIZE FOR NIBBLES ALWAYS 4 BITS
17815 :      8194      WRD_CNT = 0;
17816 :      8195
17817 :      8196      incr COUNT from 0 to 63 do      !READ 64 WORDS IN IO_BUF
17818 :      8197
17819 :      8198      incr PAT_CNT from -1 to 11 by 4 do      !READ REPEATING COUNTS OF 0-63
17820 :      8199      begin
17821 :      8200      POS_EXP = ZERO;      !FIELD SELECTOR SELECTS THE FOUR NIBBLES
17822 :      8201      PAT_INC = ONE;
17823 :      8202      TEMP = .IO_BUF [.WRD_CNT];      !GET A WORD OUT OF IO_BUF
17824 :      8203
17825 :      8204      incr CNT from 0 to 3 do      !READ THE FOUR NIBBLES IN WORD
17826 :      8205      begin
17827 :      8206
17828 :      8207      if .TEMP<.POS_EXP, .SIZ_EXP> neq (.PAT_CNT + .PAT_INC)      !COMPARE NIBBLE WITH RESPECTIVE 0-63 CNT
17829 :      8208
17830 :      8209      then
17831 :      8210      begin      !ERROR AND SET DODU_FLG IF NEQ
17832 :      8211      ERRDF (89, ARR_DAT, DUMPER);
17833 :      8212      PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_6, WRD_19);
17834 :      8213      DODU_FLG = ONE;
17835 :      8214      end;
17836 :      8215
17837 :      8216      POS_EXP = .POS_EXP + 4;      !POINT TO THE NEXT NIBBLE IN WORD
17838 :      8217      PAT_INC = .PAT_INC + 1;      !INCREMENT THE 0-63 COUNT
17839 :      8218      end;
17840 :      8219
17841 :      8220      WRD_CNT = .WRD_CNT + 1;      !GET THE NEXT IO_BUF WORD
17842 :      8221      end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (87)

```

17844 :ML4
17845 :
17846 :
17847 :      8222
17848 :      8223 ENDSUB:
17849 :      8224
17850 :      8225 if .DODU_FLG IS_SEI
17851 :      8226 then
17852 :      8227   begin
17853 :      8228     DODU (.ML_LUN);
17854 :      8229     DOCLN;
17855 :      8230   end;
17856 :      8231
17857 :      8232 ENDTST:
17861
17865 063152 004167 120740      $T37: JSR   R1,$SAVES          ;
17866 063156 162706 000012      SUB   #12,SP          ;
17867 063162 152777 000040 130024  BISH  #40,@ML.REG+40  ;
17868 063170 016705 130406      MOV   ML,DUT,R5
17869 063174 042705 177770      BIC   #177770,R5
17870 063200 142777 000007 130006  BICB  #7,@ML.REG+40
17871 063206 150577 130002      BISH  R5,@ML.REG+40
17872 063212 005066 000010      CLR   10(SP)        ; DODU.FLG      8166
17873 063216 005066 000002      CLR   2(SP)        ; WRD.CNT      8167
17874 063222 005002          CLR   R2           ; COUNT      8169
17875 063224 012703 177777      1$:  MOV   #-1,R3     ; *,PAT.CNT  8171
17876 063230 016604 000002      2$:  MOV   2(SP),R4   ; WRD.CNT,*  8173
17877 063234 006304
17878 063236 062704 011600      ADD   #10,BUF,R4
17879 063242 010305      MOV   R3,R5        ; PAT.CNT,*
17880 063244 005205      INC   R5
17881 063246 042705 177760      BIC   #177760,R5
17882 063252 142714 000017      BICB  #17,(R4)
17883 063256 150514      BISH  R5,(R4)
17884 063260 010305      MOV   R3,R5        ; PAT.CNT,*      8174
17885 063262 062705 000002      ADD   #2,R5
17886 063266 006305      ASL   R5
17887 063270 006305      ASL   R5
17888 063272 006305      ASL   R5
17889 063274 006305      ASL   R5
17890 063276 042705 177417      BIC   #177417,R5
17891 063302 142714 000360      BICB  #360,(R4)
17892 063306 150514      BISH  R5,(R4)
17893 063310 010305      MOV   R3,R5        ; PAT.CNT,*      8175
17894 063312 062705 000003      ADD   #3,R5
17895 063316 000305      SWAB  R5
17896 063320 042705 170377      BIC   #170377,R5
17897 063324 042714 007400      BIC   #7400,(R4)

```


Line	Address	Code	Label	Op	Opnd	Comments	Seq
17899							
17900							
17901							
17902	063330	050514		BIS	R5,(R4)		
17903	063332	010305		MOV	R3,R5	: PAT.CNT,*	8176
17904	063334	062705	000004	ADD	#4,R5		
17905	063340	000305		SWAB	R5		
17906	063342	006305		ASL	R5		
17907	063344	006305		ASL	R5		
17908	063346	006305		ASL	R5		
17909	063350	006305		ASL	R5		
17910	063352	042705	007777	BIC	#7777,R5		
17911	063356	042714	170000	BIC	#170000,(R4)		
17912	063362	050514		BIS	R5,(R4)		
17913	063364	005266	000002	INC	2(SP)	: WRD.CNT	8177
17914	063370	062703	000004	ADD	#4,R3	: *,PAT.CNT	8171
17915	063374	020327	000013	CMP	R3,#13	: PAT.CNT,*	
17916	063400	003713		BLE	2\$		
17917	063402	005202		INC	R2	: COUNT	8169
17918	063404	020227	000077	CMP	R2,#77	: COUNT,*	
17919	063410	003705		BLE	1\$		
17920	063412	004767	132444	JSR	PC,GD,BLK.XFER		8180
17921	063416	012777	000061	MOV	#61,@ML.REG		8181
17922	063424	105777	127574	TSTB	@ML.REG+50		
17923	063430	100375		BPL	3\$		
17924	063432	104402		TRAP	2		8182
17925	063434	005003		CLR	R3	: IO.CNT	8185
17926	063436	010304		MOV	R3,R4	: IO.CNT,*	8186
17927	063440	006304		ASL	R4		
17928	063442	005064	011600	CLR	IO.BUF(R4)		
17929	063446	005203		INC	R3	: IO.CNT	8185
17930	063450	020327	000377	CMP	R3,#377	: IO.CNT,*	
17931	063454	003770		BLE	5\$		
17932	063456	152777	000040	BISB	#40,@ML.REG+40		8186
17933	063464	016704	130112	MOV	ML,DUT,R4		
17934	063470	042704	177770	BIC	#177770,R4		
17935	063474	142777	000007	BICB	#7,@ML.REG+40		127512
17936	063502	150477	127506	BISB	R4,@ML.REG+40		
17937	063506	004767	132350	JSR	PC,GD,BLK.XFER		8189
17938	063512	012777	000071	MOV	#71,@ML.REG		8190
17939	063520	105777	127500	TSTB	@ML.REG+50		
17940	063524	100375		BPL	6\$		
17941	063526	152777	000040	BISB	#40,@ML.REG+40		8191
17942	063534	016704	130042	MOV	ML,DUT,R4		
17943	063540	042704	177770	BIC	#177770,R4		
17944	063544	142777	000007	BICB	#7,@ML.REG+40		127442
17945	063552	150477	127436	BISB	R4,@ML.REG+40		
17946	063556	012766	000004	MOV	#4,6(SP)	: *,SIZ.EXP	8193
17947	063564	005066	000002	CLR	2(SP)	: WRD.CNT	8194
17948	063570	005001		CLR	R1	: COUNT	8196
17949	063572	012703	177777	MOV	#-1,R3	: *,PAT.CNT	8198
17950	063576	005005		CLR	R5	: POS.EXP	8200
17951	063600	012716	000001	MOV	#1,(SP)	: *,PAT.INC	8201
17952	063604	015604	000002	MOV	2(SP),R4	: WRD.CNT,*	8202
17953	063610	066304		ASL	R4		

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

Address	OpCode	Operand 1	Operand 2	Operand 3	Instruction	Comments	Line No.
17955							
17956							
17957							
17958	063612	016466	011600	000004	MOV IO.BUF(R4),4(SP)	: *,TEMP	
17959	063620	005002			CLR R2	: CNT	8204
17960	063622	016646	000004	9\$:	MOV 4(SP),-(SP)	: TEMP,*	8207
17961	063626	010546			MOV R5,-(SP)	: POS.EXP,*	
17962	063630	016646	000012		MOV 12(SP),-(SP)	: SIZ.EXP,*	
17963	063634	005046			CLR -(SP)		
17964	063636	004767	117154		JSR PC,BLSGT1		
17965	063642	062706	000010		ADD #10,SP		
17966	063646	010304			MOV R3,R4	: PAT.CNT,*	
17967	063650	061604			ADD (SP),R4	: PAT.INC,*	
17968	063652	020004			CMP R0,R4		
17969	063654	001433			BEQ 10\$		
17970	063656	104455			TRAP 55	:	8211
17971	063660	000131			.WORD 131		
17972	063662	010570			.WORD ARR.DAT		
17973	063664	024052			.WORD DUMPER		
17974	063666	012746	006506		MOV #WRD.19,-(SP)	:	8212
17975	063672	012746	010004		MOV #FNC.6,-(SP)		
17976	063676	012746	006426		MOV #WRD.12,-(SP)		
17977	063702	012746	006406		MOV #WRD.10,-(SP)		
17978	063706	012746	007046		MOV #WRD.50,-(SP)		
17979	063712	012746	010172		MOV #FNC.18,-(SP)		
17980	063716	012746	006100		MOV #SIX.FMT,-(SP)		
17981	063722	012746	000007		MOV #7,-(SP)		
17982	063726	010600			MOV SP,R0	: SP,*	
17983	063730	104414			TRAP 14		
17984	063732	012766	000001	000030	MOV #1,30(SP)	: *,DODU.FLG	8213
17985	063740	062706	000020		ADD #20,SP	:	8210
17986	063744	062705	000004	10\$:	ADD #4,R5	: *,POS.EXP	8216
17987	063750	005216			INC (SP)	: PAT.INC	8217
17988	063752	005202			INC R2	: CNT	8204
17989	063754	020227	000003		CMP R2,#3	: CNT,*	
17990	063760	003720			BLE 9\$		
17991	063762	005266	000002		INC 2(SP)	: WRD.CNT	8220
17992	063766	062703	000004		ADD #4,R3	: *,PAT.CNT	8198
17993	063772	020327	000013		CMP R3,#13	: PAT.CNT,*	
17994	063776	003677			BLE 8\$		
17995	064000	005201			INC R1	: COUNT	8196
17996	064002	020127	000077		CMP R1,#77	: COUNT,*	
17997	064006	003671			BLE 7\$		
17998	064010	104467			TRAP 67	:	8221
17999	064012	006000			ROR R0		
18000	064014	103606			BLO 4\$		
18001	064016	026627	000010	000001	CMP 10(SP),#1	: DODU.FLG,*	8225
18002	064024	001004			BNE 11\$		
18003	064026	016700	127546		MOV ML.LUN,R0	:	8228
18004	064032	104451			TRAP 51		
18005	064034	104444			TRAP 44		
18006	064036	062706	000012	11\$:	ADD #12,SP	:	8116
18007	064042	000207			RTS PC		
18008							
18009							

; Routine Size: 221 words

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
18011      :ML4
18012      :
18013
18014      ; Maximum stack depth per invocation: 19 words
18019
18020
18024
18028 064044 T37::
18029 064044 004767 177102 1$: JSR PC,$T37
18030 064050 104466 TRAP 66 ;
18031 064052 00600v ROR R0
18032 064054 103773 BLO 1$
18033 064056 000207 RTS PC
18034
18035      ; Routine Size: 6 words
18036      ; Maximum stack depth per invocation: 0 words
18041
18042
18043 ;      8233 !<BLF/PAGE>
```

8230

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (88)

18045 :ML4
18046 :
18047 :
18048 :
18049 :
18050 :
18051 :
18052 :
18053 :
18054 :
18055 :
18056 :
18057 :
18058 :
18059 :
18060 :
18061 :
18062 :
18063 :
18064 :
18065 :
18066 :
18067 :
18068 :
18069 :
18070 :
18071 :
18072 :
18073 :
18074 :
18075 :
18076 :
18077 :
18078 :
18079 :
18080 :
18081 :
18082 :
18083 :
18084 :
18085 :
18086 :
18087 :
18088 :
18089 :
18090 :
18091 :
18092 :
18093 :
18094 :
18095 :
18096 :
18097 :
18098 :
18099 :

8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249
8250
8251
8252
8253
8254
8255
8256
8257
8258
8259
8260
8261
8262
8263
8264
8265
8266
8267
8268
8269
8270
8271
8272
8273
8274
8275
8276
8277
8278
8279
8280
8281
8282
8283
8284
8285

BGNTST;

!++

TEST NUMBER: TST 38

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.

IO BUF
A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READS AND WRITE
FUNCTION ARE FOUND.

STACK
A VECTOR OF 198 BYTE LOCATIONS
WHERE GOOD NIBBLE DATA IS STORED
WHEN STRIPPING AWAY BAD NIBBLE
LOCATIONS OF A BLOCK.

18101 :ML4
18102 :
18103 :
18104 :
18105 :
18106 :
18107 :
18108 :
18109 :
18110 :
18111 :
18112 :
18113 :
18114 :
18115 :
18116 :
18117 :
18118 :
18119 :
18120 :
18121 :
18122 :
18123 :
18124 :
18125 :
18126 :
18127 :
18128 :
18129 :
18130 :
18131 :
18132 :
18133 :
18134 :
18135 :
18136 :
18137 :
18138 :
18139 :
18140 :
18141 :
18142 :
18143 :
18144 :
18145 :
18146 :
18147 :
18148 :
18149 :
18150 :
18151 :
18152 :
18153 :
18154 :
18155 :

```

8286  !--
8287
8288  local
8289    SAV_NIB,
8290    DODU_FLG,
8291    NIB_BIT,
8292    TST_PAT,
8293    ALL_ONES_1,
8294    ALL_ONES_2,
8295    STK_PTR,
8296    COUNT;
8297
8298  BGNSUB;
8299  CLR_MBUS;
8300  DODU_FLG = ZERO;
8301  TST_PAT = ONE;
8302
8303  incr CNT from 0 to 15 do
8304    begin
8305    IO_BUF [.CNT] = not .TST_PAT;
8306    TST_PAT = .TST_PAT^ONE;
8307    end;
8308
8309  GD_BLK_XFER ();
8310  MLCS1 = write;
8311  TIME_OUT_LOOP;
8312  STRIPPER(21, 8);
8313  STK_PTR = -1;
8314  NIB_BIT = ONE;
8315  ALL_ONES_1 = ZERO;
8316  ALL_ONES_2 = 3;
8317
8318  incr BY_FOUR_WRDS from 0 to 3 do
8319    begin
8320
8321    incr BY_ONE_WRD from 0 to 3 do
8322      begin
8323        COUNT = ZERO;
8324
8325        until .COUNT eql .ALL_ONES_1 do
8326          begin
8327            COUNT = .COUNT + 1;
8328            STK_PTR = .STK_PTR + 1;
8329
8330            if (.stack [.STK_PTR]) neq %0'000017'
8331              then
8332                begin
8333                  ERRDF (90, SYNC, DUMPER);
8334                  PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
8335                  PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STR_PTR);
8336                  DODU_FLG = ONE;
8337                end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (88)

```

!STORES THE SHIFTED BIT PATTERN
!DROP UNIT FLAG
!NIBBLE PATTERN
!TEST PATTERN
!HOW MANY '17' NIBBLE PATTERN POSITION TO READ
!HOW MANY '17' NIBBLE PATTERN POSITION TO READ
!STACK POINTER
!COUNTER

!ONE IN A FIELD OF ZEROES

!WRITE 16 WORDS WITH SHIFTING 0 IN FIELD OF 1'S.

!SET UP A GOOD BLOCK XFERR
!WRITE SHIFTING PATTERN THROUGH SYNC BUS

!CALL ROUTINE TO STRIP OUT BAD NIBBLE DATA FROM ARRAY WORDS
!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

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (88)

```

18157 :ML4
18158 :
18159 :
18160 :      8338
18161 :      8339          end;
18162 :      8340
18163 :      8341          STK_PTR = .STK_PTR + 1;          !INCREMENT THE STACK POINTER
18164 :      8342          SAV_NIB = ( not .NIB_BIT) and (%0'000017');          !GENERATE THE SHIFTED BIT
18165 :      8343
18166 :      8344          if (.stack [.STK_PTR]) neq (.SAV_NIB)          !COMPARE STACK TO SHIFTED BIT
18167 :      8345          then
18168 :      8346              begin          !ERROR AND SET DODU_FLG IF NEQ
18169 :      8347                  ERRDF (91, SYNC, DUMPER);
18170 :      8348                  PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
18171 :      8349                  PRINTB (FMT_15, .STK_PTR);
18172 :      8350                  PRINTB (FMT_5, .SAV_NIB, .stack [.STK_PTR]);
18173 :      8351                  DODU_FLG = ONE;
18174 :      8352              end;
18175 :      8353
18176 :      8354          COUNT = ZEROES;          !CLEAR COUNT
18177 :      8355
18178 :      8356          until .COUNT eql .ALL_ONES_2 do          !READ X NUMBER OF '17' NIBBLE PAT
18179 :      8357              begin
18180 :      8358                  COUNT = .COUNT + 1;          !INCREMENT COUNT
18181 :      8359                  STK_PTR = .STK_PTR + 1;          !INCREMENT STACK POINTER
18182 :      8360
18183 :      8361                  if (.stack [.STK_PTR]) neq %0'000017'          !COMPARE STACK POINTER WITH '17'
18184 :      8362                  then
18185 :      8363                      begin          !ERROR AND SET DODU_FLG IF SET
18186 :      8364                          ERRDF (92, SYNC, DUMPER);
18187 :      8365                          PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
18188 :      8366                          PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STR_PTR);
18189 :      8367                          DODU_FLG = ONE;
18190 :      8368                      end;
18191 :      8369
18192 :      8370                  end;
18193 :      8371
18194 :      8372          NIB_BIT = .NIB_BIT^ONE;          !SHIFT THE SHIFTED NIBBLE BIT
18195 :      8373          end;
18196 :      8374
18197 :      8375          NIB_BIT = ONE;          !RESET THE SHIFTED NIBBLE BIT
18198 :      8376          ALL_ONES_1 = .ALL_ONES_1 + 1;          !READ ONE MORE '17' PATTERN
18199 :      8377          ALL_ONES_2 = .ALL_ONES_2 - 1;          !READ ONE LESS '17' PATTERN
18200 :      8378          end;
18201 :      8379
18202 :      8380          ENDSUB;
18203 :      8381
18204 :      8382          if .DODU_FLG IS_SET          !DROP THIS UNIT IF DODU_FLG SET
18205 :      8383          then
18206 :      8384              begin
18207 :      8385                  DODU (.ML_LUN);
18208 :      8386                  DOCLN;
18209 :      8387              end;
18210 :      8388
18211 :      8389          ENDTST;

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (88)

18213	:	ML4							
18214	:								
18215	:								
18219	:								
18223	064060	004167	120032		ST38:	JSR	R1,\$SAVE5	:	8232
18224	064064	162706	000012			SUB	#12,SP	:	
18225	064070	104402			1S:	TRAP	2	:	8296
18226	064072	152777	000040	127114		BISB	#40,@ML.REG+40	:	8298
18227	064100	016705	127476			MOV	ML,DUT,R5	:	
18228	064104	042705	177770			BIC	#177770,R5	:	
18229	064110	142777	000007	127076		BICB	#7,@ML.REG+40	:	
18230	064116	150577	127072			BISB	R5,@ML.REG+40	:	
18231	064122	005001				CLR	R1	:	
18232	064124	012766	000001	000006		MOV	#1,6(SP)	:	DODU.FLG 8300
18233	064132	005004				CLR	R4	:	* ,TST.PAT 8301
18234	064134	010405			2S:	MOV	R4,R5	:	CNT 8303
18235	064136	006305				ASL	R5	:	CNT,* 8305
18236	064140	016665	000006	011600		MOV	6(SP),IO.BUF(R5)	:	TST.PAT,*
18237	064146	005165	011600			COM	IO.BUF(R5)	:	
18238	064152	006366	000006			ASL	6(SP)	:	TST.PAT 8306
18239	064156	005204				INC	R4	:	CNT 8303
18240	064160	020427	000017			CMP	R4,#17	:	CNT,*
18241	064164	003763				BLE	2S	:	
18242	064166	004767	131670			JSR	PC,GD.BLK.XFER	:	
18243	064172	012777	000061	126754		MOV	#61,@ML.REG	:	8309
18244	064200	105777	127020		3S:	TSTB	@ML.REG+50	:	8310
18245	064204	100375				BPL	3S	:	
18246	064206	012746	000025			MOV	#25,-(SP)	:	8312
18247	064212	012746	000010			MOV	#10,-(SP)	:	
18248	064216	004767	127772			JSR	PC,STRIPPER	:	
18249	064222	012702	177777			MOV	#-1,R2	:	* ,STK.PTR 8313
18250	064226	012766	000001	000004		MOV	#1,4(SP)	:	* ,NIB.BIT 8314
18251	064234	005066	000010			CLR	10(SP)	:	ALL.ONES.1 8315
18252	064240	012766	000003	000006		MOV	#3,6(SP)	:	* ,ALL.ONES.2 8316
18253	064246	005004				CLR	R4	:	BY.FOUR.WRDS 8318
18254	064250	005005			4S:	CLR	R5	:	BY.ONE.WRD 8321
18255	064252	005003			5S:	CLR	R3	:	COUNT 8323
18256	064254	020366	000010		6S:	CMP	R3,10(SP)	:	COUNT,ALL.ONES.1 8325
18257	064260	001455				BEQ	7S	:	
18258	064262	005203				INC	R3	:	COUNT 8327
18259	064264	005202				INC	R2	:	STK.PTR 8328
18260	064266	126227	012612	000017		CMPB	STACK(R2),#17	:	* (STK.PTR),* 8330
18261	064274	001767				BEQ	6S	:	
18262	064276	104455				TRAP	55	:	8333
18263	064300	000132				.WORD	132	:	
18264	064302	010526				.WORD	SYNC	:	
18265	064304	024052				.WORD	DUMPER	:	
18266	064306	012746	006506			MOV	#WRD.19,-(SP)	:	8334

18268										
18269										
18270										
18271	064312	012746	007774							
18272	064316	012746	006426							
18273	064322	012746	007454							
18274	064326	012746	006716							
18275	064332	012746	006544							
18276	064336	012746	006100							
18277	064342	012746	000007							
18278	064346	010600								
18279	064350	104414								
18280	064352	010216								
18281	064354	005046								
18282	064356	116216	012612							
18283	064362	012746	177777							
18284	064366	012746	004430							
18285	064372	012746	000004							
18286	064376	010600								
18287	064400	104414								
18288	064402	012701	000001							
18289	064406	062706	000030							
18290	064412	000720								
18291	064414	005202								
18292	064416	012766	000017	000014	7\$:					
18293	064424	046666	000004	000014						
18294	064432	005000								
18295	064434	156200	012612							
18296	064440	020066	000014							
18297	064444	001454								
18298	064446	104455								
18299	064450	000133								
18300	064452	010526								
18301	064454	024052								
18302	064456	012746	006506							
18303	064462	012746	007774							
18304	064466	012746	006426							
18305	064472	012746	007454							
18306	064476	012746	006716							
18307	064502	012746	006544							
18308	064506	012746	006100							
18309	064512	012746	000007							
18310	064516	010600								
18311	064520	104414								
18312	064522	010216								
18313	064524	012746	005150							
18314	064530	012746	000002							
18315	064534	010600								
18316	064536	104414								
18317	064540	005016								
18318	064542	116216	012612							
18319	064546	016646	000040							
18320	064552	012746	004430							
18321	064556	012746	000003							
18322	064562	010600								

Address	Hex	Hex	Hex	Hex	Label	Comment	Address
18324							
18325							
18326							
18327	064564	104414			TRAP	14	
18328	064566	012701	000001		MOV	#1,R1	: *,DODU.FLG 8351
18329	064572	062706	000032		ADD	#32,SP	: 8346
18330	064576	005003		8\$:	CLR	R3	: COUNT 8354
18331	064600	020366	000006	9\$:	CMP	R3,6(SP)	: COUNT,ALL.ONES.2 8356
18332	064604	001455			BEQ	10\$	
18333	064606	005203			INC	R3	: COUNT 8358
18334	064610	005202			INC	R2	: STK.PTR 8359
18335	064612	126227	012612	000017	CMPB	STACK(R2),#17	: *(STK.PTR),* 8361
18336	064620	001767			BEQ	9\$	
18337	064622	104455			TRAP	55	: 8364
18338	064624	000134			.WORD	134	
18339	064626	010526			.WORD	SYNC	
18340	064630	024052			.WORD	DUMPER	
18341	064632	012746	006506		MOV	#WORD.19,-(SP)	: 8365
18342	064636	012746	007774		MOV	#FNC.5,-(SP)	
18343	064642	012746	006426		MOV	#WORD.12,-(SP)	
18344	064646	012746	007454		MOV	#PHR.4,-(SP)	
18345	064652	012746	006716		MOV	#WORD.39,-(SP)	
18346	064656	012746	006544		MOV	#WORD.23,-(SP)	
18347	064662	012746	006100		MOV	#SIX.FMT,-(SP)	
18348	064666	012746	000007		MOV	#7,-(SP)	
18349	064672	010600			MOV	SP,R0	: SP,*
18350	064674	104414			TRAP	14	
18351	064676	010216			MOV	R2,(SP)	: STK.PTR,* 8366
18352	064700	005046			CLR	-(SP)	
18353	064702	1 6216	012612		MOVB	STACK(R2),(SP)	: *(STK.PTR),* 8367
18354	064706	012746	177777		MOV	#-1,-(SP)	: 8363
18355	064712	012746	004430		MOV	#FMT.5,-(SP)	: 8356
18356	064716	012746	000004		MOV	#4,-(SP)	: 8372
18357	064722	010600			MOV	SP,R0	: SP,* 8321
18358	064724	104414			TRAP	14	
18359	064726	012701	000001		MOV	#1,R1	: *,DODU.FLG 8367
18360	064732	062706	000030		ADD	#30,SP	: 8363
18361	064736	000720			BR	9\$: 8356
18362	064740	006366	000004	10\$:	ASL	4(SP)	: NIB.BIT 8372
18363	064744	005205			INC	R5	: BY.ONE.WRD 8321
18364	064746	020527	000003		CMP	R5,#3	: BY.ONE.WRD,*
18365	064752	003002			BGT	11\$	
18366	064754	000167	177272		JMP	5\$	
18367	064760	012766	000001	000004	MOV	#1,4(SP)	: *,NIB.BIT 8375
18368	064766	005266	000010		INC	10(SP)	: ALL.ONES.1 8376
18369	064772	005366	000006		DEC	6(SP)	: ALL.ONES.2 8377
18370	064776	005204			INC	R4	: BY.FOUR.WRDS 8318
18371	065000	020427	000003		CMP	R4,#3	: BY.FOUR.WRDS,*
18372	065004	003002			BGT	12\$	
18373	065006	000167	177236		JMP	4\$	
18374	065012	022626		12\$:	CMP	(SP)+,(SP)+	: 8296
18375	065014	104467			TRAP	67	: 8378
18376	065016	006000			ROR	R0	
18377	065020	103002			BHIS	13\$	
18378	065022	000167	177042		JMP	1\$	

```
18380 ;ML4
18381 ;
18382 ;
18383 065026 005301 13$: DEC R1 ; DODU.FLG
18384 065030 001004 BNE 14$
18385 065032 016700 126542 MOV ML.LUN,R0 ;
18386 065036 104451 TRAP 51
18387 065040 104444 TRAP 44
18388 065042 062706 000012 14$: ADD #12,SP ;
18389 065046 000207 RTS PC ;
18390
18391 ; Routine Size: 252 words
18392 ; Maximum stack depth per invocation: 26 words
18397
18398
18402
18406 065050 T38::
18407 065050 004767 177004 1$: JSR PC,$T38 ;
18408 065054 104466 TRAP 66
18409 065056 006000 ROR R0
18410 065060 103773 BLO 1$
18411 065062 000207 RTS PC
18412
18413 ; Routine Size: 6 words
18414 ; Maximum stack depth per invocation: 0 words
18419
18420
18421 ; 8390 !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

8382

8385

8232

8387

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (89)

```

18423 :ML4
18424 :
18425 :
18426 :      8391 !
18427 :      8392 !BGNTST;
18428 :      8393 !
18429 :      8394 !++
18430 :      8395 ! TEST NUMBER: TST 39
18431 :      8396 !
18432 :      8397 ! TEST NAME: SYNC DATA BUS BIT UNIQUENESS TEST/READ PATH
18433 :      8398 !
18434 :      8399 ! TEST DESCRIPTION:
18435 :      8400 ! TEST SYNCHRONOUS DATA BUS READ
18436 :      8401 ! PATH FOR DATA BIT UNIQUENESS BY:
18437 :      8402 !
18438 :      8403 ! 1. LOADING THE FIRST 16 WORDS IN THE IO_BUF WITH A SHIFTING
18439 :      8404 ! ZERO IN A FIELD OF ONES PATTERN.
18440 :      8405 !
18441 :      8406 ! 2. VIA MBUS WRITE FUNCTION WRITE SHIFTING PATTERN INTO THE GOOD BLOCK.
18442 :      8407 !
18443 :      8408 ! 3. CLEAR THE IO_BUF.
18444 :      8409 !
18445 :      8410 ! 4. VIA MBUS READ FUNCTION OF THE SHIFTING PATTERN THROUGH THE
18446 :      8411 ! READ PATH.
18447 :      8412 !
18448 :      8413 ! 5. INTERIGATE THE IO_BUF FOR THE SHIFTING PATTERN.
18449 :      8414 !
18450 :      8415 ! IMPLICIT INPUTS:
18451 :      8416 ! IO_BUF
18452 :      8417 ! A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTION ARD FOUND.
18453 :      8418 !
18454 :      8419 !
18455 :      8420 local
18456 :      8421 DODU_FLG, !DROP UNIT FLAG
18457 :      8422 TST_PAT; !TEST PATTERN
18458 :      8423
18459 :      8424 CLR MBUS;
18460 :      8425 DODU_FLG = ZERO;
18461 :      8426 TST_PAT = ONE; !ONE IN A FIELD OF ZEROES
18462 :      8427
18463 :      8428 incr CNT from 0 to 15 do !WRITE 16 WORDS WITH SHIFTED 0 IN A FIELD OF 1'S
18464 :      8429 begin
18465 :      8430 IO_BUF [.CNT] = not .TST_PAT;
18466 :      8431 TST_PAT = .TST_PAT^ONE;
18467 :      8432 end;
18468 :      8433
18469 :      8434 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
18470 :      8435 MLCS1 = write; !WRITE SHIFTING PATTERN
18471 :      8436 TIME_OUT_LOOP;
18472 :      8437 BGNSOB;
18473 :      8438
18474 :      8439 incr CNT from 0 to 15 do !CLEAR OUT THE IO_BUF
18475 :      8440 IO_BUF [.CNT] = ZEROES;
18476 :      8441
18477 :      8442 CLR_MBUS;

```

```

18479 :ML4
18480 :
18481 :
18482 :      8443 GD_BLK_XFER ();
18483 :      8444 MLCS1 = read;
18484 :      8445 TIME_OUT_LOOP;
18485 :      8446 TST_PAT = ONE;
18486 :      8447
18487 :      8448 incr CNT from 0 to 15 do
18488 :      8449      begin
18489 :      8450
18490 :      8451      if .IO_BUF [.CNT] neq ( not .TST_PAT)
18491 :      8452      then
18492 :      8453      begin
18493 :      8454      ERRDF (93, SYNC, DUMPER);
18494 :      8455      PRINTB (FIV_FMT, WRD_23, FNC_6, WRD_19, WRD_39, PHR_4);
18495 :      8456      PRINTB (FMT_2, ( not .TST_PAT), .IO_BUF [.CNT], ( not .TST_PAT xor .IO_BUF [.CNT]));
18496 :      8457      DODU_FLG = ONE;
18497 :      8458      end;
18498 :      8459
18499 :      8460      TST_PAT = .TST_PAT^ONE;
18500 :      8461      end;
18501 :      8462
18502 :      8463 ENDSUB;
18503 :      8464
18504 :      8465 if .DODU_FLG IS_SET
18505 :      8466 then
18506 :      8467      begin
18507 :      8468      DODU (.ML_LUN);
18508 :      8469      DOCLN;
18509 :      8470      end;
18510 :      8471
18511 :      8472 ENDTST;
18515 :
18519 065064 004167 117026 $T39: JSR R1,SSAVES ;
18520 065070 152777 000040 126116 BLSB #40,@ML.REG+40 ;
18521 065076 016705 126500 MOV ML.DUT,R5 ;
18522 065102 042705 177770 BIC #177770,R5 ;
18523 065106 142777 000007 126100 BICB #7,@ML.REG+40 ;
18524 065114 150577 126074 BLSB R5,@ML.REG+40 ;
18525 065120 005005 CLR R5 ; DODU.FLG 8425
18526 065122 012704 000001 MOV #1,R4 ; *.TST.PAT 8426
18527 065126 005000 CLR R0 ; CNT 8428
18528 065130 010001 1$: MOV R0,R1 ; CNT,* 8430
18529 065132 006301 ASL R1 ;
18530 065134 010461 011600 MOV R4,IO.BUF(R1) ; TST.PAT,*
18531 065140 005161 011600 COM IO.BUF(R1) ;
18532 065144 006304 ASL R4 ; TST.PAT 8431

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (89)

!SET UP A GOOD BLOCK XFERR
!READ SHIFTING PATTERN THROUGH SYNC BUS
!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

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

18590      ;ML4
18591      ;
18592
18593 065410 011346      MOV      (R3),-(SP)
18594 065412 010146      MOV      R1, -(SP)
18595 065414 012746 004266  MOV      #FMT.2, -(SP)
18596 065420 012746 000004  MOV      #4, -(SP)
18597 065424 010600      MOV      SP, R0      ; SP,*
18598 065426 104414      TRAP     14
18599 065430 012705 000001  MOV      #1, R5      ; *,DODU.FLG
18600 065434 062706 000026  ADD      #26, SP      ;
18601 065440 006304      7$: ASL      R4      ; TST.PAT
18602 065442 005202      INC      R2      ; CNT
18603 065444 020227 000017  CMP      R2, #17      ; CNT,*
18604 065450 003713      BLE     6$
18605 065452 104467      TRAP     67
18606 065454 006000      ROR     R0      ;
18607 065456 103647      BLO     3$
18608 065460 005305      DEC     R5      ; DODU.FLG
18609 065462 001004      BNE     8$
18610 065464 016700 126110  MOV      ML.LUN, R0      ;
18611 065470 104451      TRAP     51
18612 065472 104444      TRAP     44
18613 065474 000207      8$: RTS      PC      ;
18614
18615      ; Routine Size: 133 words
18616      ; Maximum stack depth per invocation: 17 words
18621
18622
18626
18630 065476      T39::
18631 065476 004767 177362  1$: JSR      PC, $T39      ;
18632 065502 104466      TRAP     66
18633 065504 006000      ROR     R0
18634 065506 103773      BLO     1$
18635 065510 000207      RTS      PC
18636
18637      ; Routine Size: 6 words
18638      ; Maximum stack depth per invocation: 0 words
18643 ;      8473 !<BLF/PAGE>

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (90)

18645 :ML4
18646 :
18647 :
18648 :
18649 :
18650 :
18651 :
18652 :
18653 :
18654 :
18655 :
18656 :
18657 :
18658 :
18659 :
18660 :
18661 :
18662 :
18663 :
18664 :
18665 :
18666 :
18667 :
18668 :
18669 :
18670 :
18671 :
18672 :
18673 :
18674 :
18675 :
18676 :
18677 :
18678 :
18679 :
18680 :
18681 :
18682 :
18683 :
18684 :
18685 :
18686 :
18687 :
18688 :
18689 :
18690 :
18691 :
18692 :
18693 :
18694 :
18695 :
18696 :
18697 :
18698 :
18699 :

8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484
8485
8486
8487
8488
8489
8490
8491
8492
8493
8494
8495
8496
8497
8498
8499
8500
8501
8502
8503
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525

```

!
!
! BGNSTST;
!
! ++
! TEST NUMBER: TST 40
! TEST NAME: ARRAY ADDRESS MUX TEST
! TEST DESCRIPTION:
! TEST FOR UNIQUE MOS RAM ROW
! AND COLUMN ADDRESSING BY:
! 1. FIRST FINDING A ERROR FREE
! 16K OR 64K CHUNK OF MEMORY.
! THIS REPRESENTS ONE ROW OF
! EITHER 16K OR 64K MOS RAMS.
! 2. WRITE A BACKGROUND OF ALL
! ONES INTO THE GOOD CHUNK
! 3. WRITE ZEROES INTO THE FIRST
! BLOCK OF THE GOOD CHUNK.
! 4. READ REMAINING BLOCKS IN
! GOOD CHUNK FOR ONES.
!
! IMPLICIT INPUTS:
!
! IO_BUF
!
! A VECTOR OF 256 WORDS
! WHERE DATA FOR MBUS
! READ AND WRITE TRANSFERS
! CAN BE FOUND.
!
! --
! local
! DSA_ADRS,
! FND_GD_CHK;
!
! IO_BUF = ONES;
! DSA_ADRS = -.RAS_INC;
! FND_GD_CHK = ZERO;
!
! do
! begin
! CLR_MBUS;
! BAI = ONE;
! ECC_DIS = ONE;

```

```

! DSA ADRS COUNTER
! FOUND GOOD 16K/64K CHUNK FLAG
!
! LOAD FIRST IO BUF WORD WITH ONES
! REST DSA COUNT
! CLEAR FLAG
!
! DO UNTIL FOUND GOOD CHUNK OR LBT
!
! SET ON FIRST IO BUF WORD
! DISABLE ECC

```

```

18701 :ML4
18702 :
18703 :
18704 :      8526      DSA_ADRS = .DSA_ADRS + .RAS_INC;      !INCREMENT DSA_ADRS COUNTER
18705 :      8527      MLWC = .W_C_SIZE;                    !16K OR 64K WORDS
18706 :      8528      MLBA = IO_BUF;                        !LOAD UBUS_ADRS
18707 :      8529      MLDA = .DSA_ADRS;                    !LOAD DSA_ADRS
18708 :      8530      ML_FUNC = write;                      !DO A WRITE FUNCTION
18709 :      8531      TIME_OUT_LOOP;
18710 :      8532
18711 :      8533      if .SC IS_NOT_SET                      !DID XFERR CAUSE AN SC
18712 :      8534      then
18713 :      8535      begin
18714 :      8536      MLWC = .W_C_SIZE;                    !XFERR WAS OK
18715 :      8537      MLBA = IO_BUF;                        !LOAD WORD COUNT
18716 :      8538      MLDA = .DSA_ADRS;                    !LOAD UBUS_ADRS
18717 :      8539      ML_FUNC = WRT_CHK;                   !LOAD DSA_ADRS
18718 :      8540      TIME_OUT_LOOP;                       !DO A WRITE CHECK FUNCTION
18719 :      8541
18720 :      8542      if .SC IS_NOT_SET                      !IS THIS CHUNK GOOD
18721 :      8543      then
18722 :      8544      FND_GD_CHK = ONE;                     !YES SET FLG
18723 :      8545
18724 :      8546      end
18725 :      8547
18726 :      8548      end
18727 :      8549      until (.FND_GD_CHK IS_SET ) or (.DSA_ADRS eql .LST_ARR + .ARR_INC);
18728 :      8550
18729 :      8551      !REPEAT UNTIL FOUND GOOD CHUNCK OR AT LBT
18730 :      8552
18731 :      8553      if .DSA_ADRS eql .LST_ARR + .ARR_INC
18732 :      8554      then
18733 :      8555      begin
18734 :      8556      ERRDF (111, INTER, DUMPER);
18735 :      8557      PRINTB (FIV_FMT, FNC_13, FNC_17, WRD_50, WRD_60, WRD_56);
18736 :      8558      PRINTB (THR_FMT, WRD_14, PHR_10, FNC_15);
18737 :      8559      DODU (.ML_LDN);
18738 :      8560      DOCLN;
18739 :      8561      end
18740 :      8562      else
18741 :      8563      begin
18742 :      8564      CLR_MBUS;
18743 :      8565      BAI = ONE;
18744 :      8566      ECC_DIS = ONE;                          !DISABLE ECC
18745 :      8567      IO_BUF = ZEROES;                       !FIRST BLOCK IN CHUNCK GETS ZEROES
18746 :      8568      MLDA = .DSA_ADRS;                    !LOAD DSA
18747 :      8569      MLWC = not 255;                       !LOAD WORD COUNT
18748 :      8570      MLBA = IO_BUF;                        !LOAD UBUS_ADRS
18749 :      8571      ML_FUNC = write;                      !DO A WRITE FUNCTION
18750 :      8572      TIME_OUT_LOOP;
18751 :      8573      CLR_MBUS;
18752 :      8574      BAI = ONE;
18753 :      8575      IO_BUF = ONES;
18754 :      8576      ECC_DIS = ONE;
18755 :      8577      MLDA = .DSA_ADRS + 1;

```



```

18757 :ML4
18758 :
18759 :
18760 : 8578 MLBA = IO_BUF;
18761 : 8579 MLWC = .W.C.SIZE + 256;
18762 : 8580 ML_FUNC = WRT_CHK;
18763 : 8581 TIME_OUT_LOOP;
18764 : 8582
18765 : 8583 if .WCE IS_SET
18766 : 8584 then
18767 : 8585 begin
18768 : 8586 ERRDF (112, ASYNC, DUMPER);
18769 : 8587 PRINTB (FOR_FMT, FNC 17, WRD_50, WRD_60, WRD_14);
18770 : 8588 PRINTB (FMT_9, (.MLDA - 1));
18771 : 8589 DODU (.ML_LDN);
18772 : 8590 DOCLN;
18773 : 8591 end;
18774 : 8592
18775 : 8593 end;
18776 : 8594
18777 : 8595 ENDTST;
18781
18785 065512 004167 116326 $T40: JSR R1,$SAVE2
18786 065516 012767 177777 124054 MOV #1,IO_BUF
18787 065524 016702 125374 MOV RAS.INC,R2
18788 065530 005402 NEG R2
18789 065532 005000 CLR R0
18790 065534 152777 000040 125452 1$: BISB #40,@ML.REG+40
18791 065542 016701 126034 MOV ML.DUT,R1
18792 065546 042701 177770 BIC #177770,R1
18793 065552 142777 000007 125434 BICB #7,@ML.REG+40
18794 065560 150177 125430 BISB R1,@ML.REG+40
18795 065564 152777 000010 125422 BISB #10,@ML.REG+40
18796 065572 152777 000002 125474 BISB #2,@ML.REG+120
18797 065600 066702 125320 ADD RAS.INC,R2
18798 065604 016777 125312 125352 MOV W.C.SIZE,@ML.REG+10
18799 065612 012777 011600 125354 MOV #10.BUF,@ML.REG+20
18800 065620 010277 125360 MOV R2,@ML.REG+30
18801 065624 142777 000077 125322 BICB #77,@ML.REG
18802 065632 152777 000061 125314 BISB #61,@ML.REG
18803 065640 105777 125360 2$: TSTB @ML.REG+50
18804 065644 100375 BPL 2$
18805 065646 032777 100000 125300 BIT #100000,@ML.REG
18806 065654 001027 BNE 4$
18807 065656 016777 125240 125300 MOV W.C.SIZE,@ML.REG+10
18808 065664 012777 011600 125302 MOV #10.BUF,@ML.REG+20
18809 065672 010277 125306 MOV R2,@ML.REG+30
18810 065676 142777 000077 125250 BICB #77,@ML.REG
    
```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
 22-Dec-1980 09:21 2 PA:<NEALE>BL3ML4.BLI.2 (90)

!LOAD UBUS ADRS
 !LOAD WORD COUNT WITH LESS ONE BLOCK
 !DO A WRITE CHECK FUNCTION
 !WERE ANY BITS DISTURBED
 !ERROR IF WRITE CHECK FOUND BAD DATA

8472
 8517
 8518
 8519
 8522
 8524
 8525
 8526
 8527
 8528
 8529
 8530
 8533
 8536
 8537
 8538
 8539

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 412 J 15

SEQ 0399

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
18812 ;ML4
18813 ;
18814 ;
18815 065704 152777 000051 125242 BISB #51,@ML.REG
18816 065712 105777 125306 3S: TSTB @ML.REG+50
18817 065716 100375 BPL 3S
18818 065720 032777 100000 125226 BIT #100000,@ML.REG ; 8542
18819 065726 001002 BNE 4S
18820 065730 012700 000001 MOV #1,R0 ; *,FND.GD.CHK
18821 065734 020027 000001 4S: CMP R0,#1 ; FND.GD.CHK,* 8544
18822 065740 001406 BEQ 5S ; 8549
18823 065742 016701 123630 MOV LST.ARR,R1
18824 065746 066701 123610 ADD ARR.INC,R1
18825 065752 020201 CMP R2,R1 ; DSA.ADRS,*
18826 065754 001267 BNE 1S
18827 065756 016701 123614 5S: MOV LST.ARR,R1 ; 8553
18828 065762 066701 123574 ADD ARR.INC,R1
18829 065766 020201 CMP R2,R1 ; DSA.ADRS,*
18830 065770 001047 BNE 6S
18831 065772 104455 TRAP 5S ; 8556
18832 065774 000157 .WORD 157
18833 065776 010672 .WORD INTER
18834 066000 024052 .WORD DUMPER
18835 066002 012746 007122 MOV @WORD.56,-(SP) ; 8557
18836 066006 012746 007156 MOV @WORD.60,-(SP)
18837 066012 012746 007046 MOV @WORD.50,-(SP)
18838 066016 012746 010162 MOV #FNC.17,-(SP)
18839 066022 012746 010112 MOV #FNC.13,-(SP)
18840 066026 012746 006062 MOV #FIV.FMT,-(SP)
18841 066032 012746 000006 MOV #6,-(SP)
18842 066036 010600 MOV SP,R0 ; SP,*
18843 066040 104414 TRAP 14
18844 066042 012716 010136 MOV #FNC.15,(SP) ; 8558
18845 066046 012746 007564 MOV #PHR.10,-(SP)
18846 066052 012746 006442 MOV @WORD.14,-(SP)
18847 066056 012746 006034 MOV #THR.FMT,-(SP)
18848 066062 012746 000004 MOV #4,-(SP)
18849 066066 010600 MOV SP,R0 ; SP,*
18850 066070 104414 TRAP 14
18851 066072 016700 125502 MOV ML.LUN,R0 ; 8559
18852 066076 104451 TRAP 51
18853 066100 104444 TRAP 44
18854 066102 062706 000026 ADD #26,SP ; 8555
18855 066106 000207 RTS PC ; 8553
18856 066110 152777 000040 125076 6S: BISB #40,@ML.REG+40 ; 8563
18857 066116 016701 125460 MOV ML.DUT,R1
18858 066122 042701 177770 BIC #177770,R1
18859 066126 142777 000007 125060 BICB #7,@ML.REG+40
18860 066134 150177 125054 BISB R1,@ML.REG+40
18861 066140 152777 000010 125046 BISB #10,@ML.REG+40 ; 8565
18862 066146 152777 000002 125120 BISB #2,@ML.REG+120 ; 8566
18863 066154 005067 123420 CLR IO.BUF ; 8567
18864 066160 010277 125020 MOV R2,@ML.REG+30 ; DSA.ADRS,* 8568
18865 066164 012777 177400 124772 MOV #-400,@ML.REG+10 ; 8569
18866 066172 012777 011600 124774 MOV #IO.BUF,@ML.REG+20 ; 8570
```

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

```

18868      ;ML4
18869      ;
18870
18871 066200 142777 000077 124746      BICB      #77,@ML.REG      ;
18872 066206 152777 000061 124740      BISB      #61,@ML.REG      ;
18873 066214 105777 125004      7$:      TSTB      @ML.REG+50
18874 066220 100375      BPL      7$
18875 066222 152777 000040 124764      BISB      #40,@ML.REG+40      ;
18876 066230 016701 125346      MOV      ML,DUT,R1
18877 066234 042701 177770      BIC      #177770,R1
18878 066240 142777 000007 124746      BICB      #7,@ML.REG+40
18879 066246 150177 124742      BISB      R1,@ML.REG+40
18880 066252 152777 000010 124734      BISB      #10,@ML.REG+40      ;
18881 066260 012767 177777 123312      MOV      #-1,IO.BUF      ;
18882 066266 152777 000002 125000      BISB      #2,@ML.REG+120      ;
18883 066274 010201      MOV      R2,R1      ; DSA.ADRS,*
18884 066276 005201      INC      R1
18885 066300 010177 124700      MOV      R1,@ML.REG+30
18886 066304 012777 011600 124662      MOV      #IO.BUF,@ML.REG+20      ;
18887 066312 016702 124604      MOV      W.C.SIZE,R2      ;
18888 066316 062702 000400      ADD      #400,R2
18889 066322 010277 124636      MOV      R2,@ML.REG+10
18890 066326 142777 000077 124620      BICB      #77,@ML.REG      ;
18891 066334 152777 000051 124612      BISB      #51,@ML.REG
18892 066342 105777 124656      8$:      TSTB      @ML.REG+50
18893 066346 100375      BPL      8$
18894 066350 032777 040000 124636      BIT      #40000,@ML.REG+40      ;
18895 066356 001441      BEQ      9$
18896 066360 104455      TRAP     55      ;
18897 066362 000160      .WORD   160
18898 066364 010464      .WORD   ASYNC
18899 066366 024052      .WORD   DUMPER
18900 066370 012746 006442      MOV      #WORD.14,-(SP)      ;
18901 066374 012746 007156      MOV      #WORD.60,-(SP)
18902 066400 012746 007046      MOV      #WORD.50,-(SP)
18903 066404 012746 010162      MOV      #FNC.17,-(SP)
18904 066410 012746 006046      MOV      #FOR.FMT,-(SP)
18905 066414 012746 000005      MOV      #5,-(SP)
18906 066420 010600      MOV      SP,R0      ; SP,*
18907 066422 104414      TRAP     14
18908 066424 017716 124554      MOV      @ML.REG+30,(SP)      ;
18909 066430 005316      DEC      (SP)
18910 066432 012746 004644      MOV      #FMT.9,-(SP)
18911 066436 012746 000002      MOV      #2,-(SP)
18912 066442 010600      MOV      SP,R0      ; SP,*
18913 066444 104414      TRAP     14
18914 066446 016700 125126      MOV      ML.LUN,R0      ;
18915 066452 104451      TRAP     51
18916 066454 104444      TRAP     44
18917 066456 062706 000020      ADD      #20,SP
18918 066462 000207      9$:      RTS      PC
18919
18920      ; Routine Size: 245 words
18921      ; Maximum stack depth per invocation: 14 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

8571
8572
8574
8575
8576
8577
8578
8579
8580
8583
8586
8587
8588
8589
8585
8472

18930
18931
18935
18939 066464
18940 066464 004767 177022
18941 066470 104466
18942 066472 006000
18943 066474 103773
18944 066476 000207
18945
18946
18947
18952
18953
18954 ; 8596 !<BLF/PAGE>

T40::
1\$: JSR PC,\$T40 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC
; Routine Size: 6 words
; Maximum stack depth per invocation: 0 words

8593

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (91)

```

18956 :ML4
18957 :
18958 :
18959 :      8597 !
18960 :      8598 !BGNTST;
18961 :      8599 !
18962 :      8600 !++
18963 :      8601 !TEST NUMBER: TST 41
18964 :      8602 !
18965 :      8603 !TEST NAME: NIBBLE OFFSET TEST
18966 :      8604 !
18967 :      8605 !TEST DESCRIPTION:
18968 :      8606 !TEST NIBBLE OFFSET COUNTERS TO OFFSET GOOD NIBBLE DATA A MAX
18969 :      8607 !OF 14 WORDS ON DETECTION OF ALL BAD NIBBLES BY:
18970 :      8608 !
18971 :      8609 !1. LOADING FIRST 2 1/4 WORDS OF THE IO_BUF WITH ZEROES AND THE
18972 :      8610 !REMAINING OF BUFFER WITH ONES.
18973 :      8611 !
18974 :      8612 !2. VIA DAT_DM MODE WRITE THE GOOD BLOCK WITH BACKGROUND
18975 :      8613 !ON ONES.
18976 :      8614 !
18977 :      8615 !3. VIA PROM R/W MODE FORCE ALL ARRAY NIBBLES BAD.
18978 :      8616 !
18979 :      8617 !4. VIA A MBUS WRITE FUNCTION LOAD IO_BUF INTO THE GOOD BLOCK.
18980 :      8618 !
18981 :      8619 !5. VIA DAT_DM MODE READ FIRST 15 ARRAY WORDS FOR ZEROES AND THE
18982 :      8620 !REMAINING WORDS FOR ONES.
18983 :      8621 !
18984 :      8622 !IMPLICIT INPUTS:
18985 :      8623 !PD_TEMP
18986 :      8624 !A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND
18987 :      8625 !ACCESSED FROM.
18988 :      8626 !
18989 :      8627 !IO_BUF
18990 :      8628 !A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
18991 :      8629 !FUNCTIONS ARE FOUND.
18992 :      8630 !
18993 :      8631 !
18994 :      8632 !local
18995 :      8633 !DODU_FLG, !DROP UNIT FLAG
18996 :      8634 !TST_PAT, !TEST PATTERN
18997 :      8635 !START, !STARTING WORD
18998 :      8636 !FINISH, !ENDING WORD
18999 :      8637 !ERR_FLG; !ERROR FLAG
19000 :      8638 !
19001 :      8639 !BGNSUB;
19002 :      8640 !CLR_MBUS;
19003 :      8641 !DODU_FLG = ZERO;
19004 :      8642 !
19005 :      8643 !incr WD_CNT from 0 to 255 do !LOAD IO_BUF WITH ONES
19006 :      8644 !IO_BUF [WD_CNT] = ONES;
19007 :      8645 !
19008 :      8646 !IO_BUF [0] = ZEROES; !LOAD FIRST 2 1/4 WORDS WITH ZEROES
19009 :      8647 !IO_BUF [1] = ZEROES;
19010 :      8648 !IO_BUF [2] = %o'177760';

```

```

19012 :ML4
19013 :
19014 :
19015 :      8649 MLD1 = ONES;
19016 :      8650 MLD2 = ONES;
19017 :      8651 MLE2 = ONES;
19018 :      8652 DAT_DM_XFER ();
19019 :      8653 MLCS1 = write;
19020 :      8654
19021 :      8655 incr WD_CNT from 0 to 127 do
19022 :      8656      begin
19023 :      8657      DELAY (ONE_US);
19024 :      8658      DAT_CLK = ONE;
19025 :      8659      end;
19026 :      8660
19027 :      8661 CLR_MBUS;
19028 :      8662 WRT_PD (ONES, 19);
19029 :      8663 PROM_RW = ONE;
19030 :      8664 GD_BLK_XFER ();
19031 :      8665 MLCS1 = write;
19032 :      8666
19033 :      8667 TIME_OUT_LOOP;
19034 :      8668 CLR_MBUS;
19035 :      8669 START = ZERO;
19036 :      8670 FINISH = 14;
19037 :      8671 TST_PAT = ZEROES;
19038 :      8672 DAT_DM_XFER ();
19039 :      8673 MLCS1 = read;
19040 :      8674 DELAY (ONE_US);
19041 :      8675
19042 :      8676 incr TWICE from 0 to 1 do
19043 :      8677      begin
19044 :      8678
19045 :      8679      incr WRD_CNT from .START to .FINISH do
19046 :      8680      begin
19047 :      8681      PD_TEMP = .MLPD;
19048 :      8682      DAT_CLK = ONE;
19049 :      8683      DELAY (ONE_US);
19050 :      8684      RD_LNG_WRD;
19051 :      8685
19052 :      8686      incr NIB_PTR from 0 to 8 do
19053 :      8687
19054 :      8688      if .PD_TEMP [.NIB_PTR] IS_NOT_SET
19055 :      8689      then
19056 :      8690      begin
19057 :      8691      TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
19058 :      8692
19059 :      8693      if .ERR_FLG IS_SET
19060 :      8694      then
19061 :      8695      begin
19062 :      8696      ERRDF (94, ARR_DAT, DUMPER);
19063 :      8697      PRINTB (THR_FMT, WRD_41, WRD_46, WRD_10);
19064 :      8698      PRINTB (FMT_6, .NIB_PTR);
19065 :      8699      DODU_FLG = ONE;
19066 :      8700      end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (91)

```

!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
!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

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
 22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (91)

```

19068 ;ML4
19069 :
19070 :
19071 :      8701
19072 :      8702          end;
19073 :      8703
19074 :      8704          end;
19075 :      8705
19076 :      8706      TST PAT = not .TST_PAT;
19077 :      8707      START = 15;
19078 :      8708      FINISH = 126;
19079 :      8709      end
19080 :      8710
19081 :      8711      ENDSUB;
19082 :      8712
19083 :      8713      if .DODU_FLG IS_SET
19084 :      8714      then
19085 :      8715          begin
19086 :      8716              DODU (.ML_LUN);
19087 :      8717              DOCLN;
19088 :      8718          end;
19089 :      8719
19090 :      8720      ENDTST;
19091 :
19092 :
19093 :
19094 :
19095 :
19096 :
19097 :
19098 066500 004167 115412      $T41:      JSR      R1, $SAVE5
19099 066504 162706 000012      SUB      #12, SP
19100 066510 104402      1$:      TRAP      2
19101 066512 152777 000040 124474      BISB     #40, @ML.REG+40
19102 066520 016704 125056      MOV      ML.DUT, R4
19103 066524 042704 177770      BIC      #177770, R4
19104 066530 142777 000007 124456      BICB     #7, @ML.REG+40
19105 066536 150477 124452      BISB     R4, @ML.REG+40
19106 066542 005066 000004      CLR      4(SP)
19107 066546 005002      CLR      R2
19108 066550 010203      2$:      MOV      R2, R3
19109 066552 006303      ASL      R3
19110 066554 012763 177777 011600      MOV      #-1, IO.BUF(R3)
19111 066562 005202      INC      R2
19112 066564 020227 000377      CMP      R2, #377
19113 066570 003767      BLE      2$
19114 066572 005067 123002      CLR      IO.BUF
19115 066576 005067 123000      CLR      IO.BUF+2
19116 066602 012767 177760 122774      MOV      #-20, IO.BUF+4
19117 066610 012777 177777 124526      MOV      #-1, @ML.REG+170
19118 066616 012777 177777 124530      MOV      #-1, @ML.REG+200
19119 066624 012777 177777 124502      MOV      #-1, @ML.REG+160
19120 066632 004767 125324      JSR      PC, DAT.DM.XFER
19121 066636 012777 000061 124310      MOV      #61, @ML.REG
    
```

!NOW READ FOR ONES
 !START A 15
 !END AT 126

!DROP THIS UNIT IF DODU_FLG SET

8595
 8637
 8639
 8641
 8643
 8644
 8643
 8646
 8647
 8648
 8649
 8650
 8651
 8652
 8653

19123						:ML4				
19124						:				
19125										
19126	066644	005001					CLR	R1	:	WD.CNT
19127	066646	012702	000001			3\$:	MOV	#1,R2	:	*,SSTMP2
19128	066652	001411				4\$:	BEQ	7\$:	
19129	066654	016703	113236				MOV	LSDLY,R3	:	*,SSTMP1
19130	066660	001404					BEQ	6\$:	
19131	066662	005066	000010			5\$:	CLR	10(SP)	:	SSTMP
19132	066666	005303					DEC	R3	:	SSTMP1
19133	066670	001374					BNE	5\$:	
19134	066672	005302				6\$:	DEC	R2	:	SSTMP2
19135	066674	000766					BR	4\$:	
19136	066676	152777	000020	124370		7\$:	BISB	#20,@ML.REG+120	:	
19137	066704	005201					INC	R1	:	WD.CNT
19138	066706	020127	000177				CMP	R1,#177	:	WD.CNT,*
19139	066712	003755					BLE	3\$:	
19140	066714	152777	000040	124272			BISB	#40,@ML.REG+40	:	
19141	066722	016704	124654				MOV	ML,DUT,R4	:	
19142	066726	042704	177770				BIC	#177770,R4	:	
19143	066732	142777	000007	124254			BICB	#7,@ML.REG+40	:	
19144	066740	150477	124250				BISB	R4,@ML.REG+40	:	
19145	066744	012746	177777				MOV	#-1,-(SP)	:	
19146	066750	012746	000023				MOV	#23,-(SP)	:	
19147	066754	004767	132240				JSR	PC,WRT.PD	:	
19148	066760	152777	000100	124306			BISB	#100,@ML.REG+120	:	
19149	066766	004767	127070				JSR	PC,GD.BLK.XFER	:	
19150	066772	012777	000061	124154			MOV	#61,@ML.REG	:	
19151	067000	105777	124220			8\$:	TSTB	@ML.REG+50	:	
19152	067004	100375					BPL	8\$:	
19153	067006	152777	000040	124200			BISB	#40,@ML.REG+40	:	
19154	067014	016704	124562				MOV	ML,DUT,R4	:	
19155	067020	042704	177770				BIC	#177770,R4	:	
19156	067024	142777	000007	124162			BICB	#7,@ML.REG+40	:	
19157	067032	150477	124156				BISB	R4,@ML.REG+40	:	
19158	067036	005066	000006				CLR	6(SP)	:	START
19159	067042	012766	000016	000004			MOV	#16,4(SP)	:	*,FINISH
19160	067050	005005					CLR	R5	:	TST.PAT
19161	067052	004767	125104				JSR	PC,DAT.DM.XFER	:	
19162	067056	012777	000071	124070			MOV	#71,@ML.REG	:	
19163	067064	012702	000001				MOV	#1,R2	:	*,SSTMP2
19164	067070	001411				9\$:	BEQ	12\$:	
19165	067072	016703	113020				MOV	LSDLY,R3	:	*,SSTMP1
19166	067076	001404					BEQ	11\$:	
19167	067100	005066	000014			10\$:	CLR	14(SP)	:	SSTMP
19168	067104	005303					DEC	R3	:	SSTMP1
19169	067106	001374					BNE	10\$:	
19170	067110	005302				11\$:	DEC	R2	:	SSTMP2
19171	067112	000766					BR	9\$:	
19172	067114	005004				12\$:	CLR	R4	:	TWICE
19173	067116	016601	000006			13\$:	MOV	6(SP),R1	:	START,WTD.CNT
19174	067122	005301					DEC	R1	:	WTD.CNT
19175	067124	000534					BR	22\$:	
19176	067126	017767	124252	123764		14\$:	MOV	@ML.REG+230,PD.TEMP	:	
19177	067134	152777	000020	124132			BISB	#20,@ML.REG+120	:	

8655
8657
8658
8659
8662
8663
8664
8665
8667
8669
8670
8671
8672
8673
8674
8676
8679
8681
8682


```

19235      ;ML4
19236      ;
19237
19238 067402 062706 000006 20$: ADD #6,SP ;
19239 067406 005202 21$: INC R2 ; NIB.PTR
19240 067410 020227 000010 CMP R2,#10 ; NIB.PTR,*
19241 067414 003700 BLE 19$ ;
19242 067416 005201 22$: INC R1 ; WRD.CNT
19243 067420 020166 000004 CMP R1,4(SP) ; WRD.CNT,FINISH
19244 067424 003640 BLE 14$ ;
19245 067426 005105 COM R5 ; TST.PAT
19246 067430 012766 000017 000006 MOV #17,6(SP) ; *,START
19247 067436 012766 000176 000004 MOV #176,4(SP) ; *,FINISH
19248 067444 005204 INC R4 ; TWICE
19249 067446 020427 000001 CMP R4,#1 ; TWICE,*
19250 067452 003621 BLE 13$ ;
19251 067454 022626 CMP (SP)+,(SP)+ ;
19252 067456 104467 TRAP 67 ;
19253 067460 006000 ROR R0 ;
19254 067462 103002 BHIS 23$ ;
19255 067464 000167 177020 JMP 1$ ;
19256 067470 026627 000004 000001 23$: CMP 4(SP),#1 ; DODU.FLG,*
19257 067476 001004 BNE 24$ ;
19258 067500 016700 124074 MOV ML.LUN,R0 ;
19259 067504 104451 TRAP 51 ;
19260 067506 104444 TRAP 44 ;
19261 067510 062706 000012 24$: ADD #12,SP ;
19262 067514 000207 RTS PC ;
19263
19264 ; Routine Size: 263 words
19265 ; Maximum stack depth per invocation: 23 words
19270
19271
19275
19279 067516 T41::
19280 067516 004767 176756 1$: JSR PC,$T41 ;
19281 067522 104466 TRAP 66 ;
19282 067524 006000 ROR R0 ;
19283 067526 103773 BLO 1$ ;
19284 067530 000207 RTS PC ;
19285
19286 ; Routine Size: 6 words
19287 ; Maximum stack depth per invocation: 0 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

8690
8686

8679

8706

8707

8708

8676

8637

8709

8713

8716

8595

8718

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 421^{F.16}

SEQ 0408

19296
19297
19298 ; 8721 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (92)

19300 :ML4
19301 :
19302 :
19303 :
19304 :
19305 :
19306 :
19307 :
19308 :
19309 :
19310 :
19311 :
19312 :
19313 :
19314 :
19315 :
19316 :
19317 :
19318 :
19319 :
19320 :
19321 :
19322 :
19323 :
19324 :
19325 :
19326 :
19327 :
19328 :
19329 :
19330 :
19331 :
19332 :
19333 :
19334 :
19335 :
19336 :
19337 :
19338 :
19339 :
19340 :
19341 :
19342 :
19343 :
19344 :
19345 :
19346 :
19347 :
19348 :
19349 :
19350 :
19351 :
19352 :
19353 :
19354 :

8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
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

```

!
BGNTST;

!++
TEST NUMBER:  TST 42
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;
IO_BUF = ONES;
GD_BLK_XFER ();
MLCS1 = write;
TIME_OUT_LOOP;

incr TWICE from 0 to 1 do
begin
BGNSUB;
CLR_MBUS;

incr CNT from 0 to 255 do
IO_BUF [.CNT] = ZEROES;

if .TWICE eql 1
then
begin
PROM_RW = ONE;
MLPD = %o'777';

```

```

!SET ON FIRST IO_BUF ADRS
!LOAD FIRST IO_BUF ADRS
!SET UP A GOOD_BLOCK XFERR
!WRITE BACKGROUND PATTERN

!FORCE CLASS 'A' AND CLASS 'B' ERRORS

!CLEAR OUT IO_BUF

!IF 2ND PASS THEN FORCE 'B' ERROR

```

19356 :ML4
19357 :
19358 :
19359 :
19360 :
19361 :
19362 :
19363 :
19364 :
19365 :
19366 :
19367 :
19368 :
19369 :
19370 :
19371 :
19372 :
19373 :
19374 :
19375 :
19376 :
19377 :
19378 :
19379 :
19380 :
19381 :
19382 :
19383 :
19384 :
19385 :
19386 :
19387 :
19388 :
19389 :
19390 :
19391 :
19392 :
19393 :
19394 :
19395 :
19396 :
19397 :
19398 :
19399 :
19400 :
19401 :
19402 :
19403 :
19404 :
19405 :
19406 :
19407 :
19408 :
19409 :
19410 :

8774
8775
8776
8777
8778
8779
8780
8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813
8814
8815
8816
8817
8818
8819
8820
8821
8822
8823
8824
8825

```

end;
GD_BLK_XFER ();
MLCS1 = read;
if .TWICE eql 0 then MLDA = ONES;
TIME_OUT_LOOP;
if .TWICE eql 0
then
begin
incr WRD_CNT from 0 to 64 do
begin
if .IO_BUF [.WRD_CNT] neq ONES
then
begin
ERRDF (95, SYNC, DUMPER);
PRINTB (FOR_FMT, PHR_8, FNC_13, WRD_19, WRD_10);
exitloop;
end;
end;
end
else
begin
incr WRD_CNT from 0 to 64 do
begin
if .IO_BUF [.WRD_CNT] neq ZEROES
then
begin
ERRDF (96, SYNC, DUMPER);
PRINTB (FOR_FMT, PHR_9, FNC_13, WRD_19, WRD_10);
exitloop;
end;
end;
end;
if .SC IS_NOT_SET
then
begin
ERRDF (104, SYNC, DUMPER);
PRINTB (FIV_FMT, WRD_59, PHR_1, WRD_11, WRD_19, FNC_13)
end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (92)

!SET UP A GOOD BLOCK XFERR
!DO A READ FUNCTION
!IF FIRST PASS THEN FORCE AN 'A' ERROR
!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
!ERROR IF NOT SET

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (92)

```

19412 :ML4
19413 :
19414 :
19415 :      8826      ENDSUB;
19416 :      8827      end;
19417 :      8828
19418 :      8829      ENDTST;
19422
19426 067532 004167 114322      $T42:  JSR      R1,$SAVE3      :
19427 067536 152777 000040 123450  BISB    #40,@ML.REG+40  :
19428 067544 016703 124032      MOV     ML.DUT,R3      :
19429 067550 042703 177770      BIC    #177770,R3     :
19430 067554 142777 000007 123432  BICB   #7,@ML.REG+40  :
19431 067562 150377 123426      BISB   R3,@ML.REG+40  :
19432 067566 152777 000010 123420  BISB   #10,@ML.REG+40 :
19433 067574 012767 177777 121776  MOV    #-1,IO.BUF     :
19434 067602 004767 126254      JSR    PC,GD.BLK.XFER :
19435 067606 012777 000061 123340  MOV    #61,@ML.REG    :
19436 067614 105777 123404      1$:   TSTB   @ML.REG+50  :
19437 067620 100375      BPL    1$             :
19438 067622 005003      CLR    R3             : TWICE
19439 067624 104402      2$:   TRAP   2             :
19440 067626 152777 000040 123360  BISB   #40,@ML.REG+40 :
19441 067634 016702 123742      MOV    ML.DUT,R2     :
19442 067640 042702 177770      BIC    #177770,R2    :
19443 067644 142777 000007 123342  BICB   #7,@ML.REG+40  :
19444 067652 150277 123336      BISB   R2,@ML.REG+40 :
19445 067656 005000      CLR    R0             : CNT
19446 067660 010001      3$:   MOV    R0,R1       : CNT,*
19447 067662 006301      ASL    R1             :
19448 067664 005061 011600      CLR    IO.BUF(R1)    :
19449 067670 005200      INC    R0             : CNT
19450 067672 020027 000377      CMP    R0,#377       : CNT,*
19451 067676 003770      BLE   3$             :
19452 067700 020327 000001      CMP    R3,#1         : TWICE,*
19453 067704 001006      BNE   4$             :
19454 067706 152777 000100 123360  BISB   #100,@ML.REG+120 :
19455 067714 012777 000777 123462  MOV    #777,@ML.REG+230 :
19456 067722 004767 126134      4$:   JSR    PC,GD.BLK.XFER :
19457 067726 012777 000071 123220  MOV    #71,@ML.REG    :
19458 067734 005002      CLR    R2             :
19459 067736 005703      TST   R3             : TWICE
19460 067740 001004      BNE   5$             :
19461 067742 005202      INC    R2             :
19462 067744 012777 177777 123232  MOV    #-1,@ML.REG+30 :
19463 067752 105777 123246      5$:   TSTB   @ML.REG+50  :
19464 067756 100375      BPL    5$             :
19465 067760 006002      ROR   R2             :

```

8720
8723

8755
8756
8757
8758

8761
8762
8763

8766
8767

8766

8769

8772
8773
8776
8777
8779

8783

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Label	Instruction	Comment	Line No.
19467						:ML4			
19468						:			
19469						:			
19470	067762	103037					BCC 8\$		
19471	067764	005002					CLR R2	: WRD.CNT	8787
19472	067766	010201				6\$:	MOV R2,R1	: WRD.CNT,*	8790
19473	067770	006301					ASL R1		
19474	067772	026127	011600	177777			CMP IO.BUF(R1),#-1		
19475	070000	001423					BEQ 7\$		
19476	070002	104455					TRAP 55	:	8793
19477	070004	000137					.WORD 137		
19478	070006	010526					.WORD SYNC		
19479	070010	024052					.WORD DUMPER		
19480	070012	012746	006406				MOV #WRD.10,-(SP)	:	8794
19481	070016	012746	006506				MOV #WRD.19,-(SP)		
19482	070022	012746	010112				MOV #FNC.13,-(SP)		
19483	070026	012746	007540				MOV #PHR.8,-(SP)		
19484	070032	012746	006046				MOV #FOR.FMT,-(SP)		
19485	070036	012746	000005				MOV #5,-(SP)		
19486	070042	010600					MOV SP,R0	: SP,*	
19487	070044	104414					TRAP 14		
19488	070046	000435					BR 10\$		8795
19489	070050	005202				7\$:	INC R2	: WRD.CNT	8787
19490	070052	020227	000100				CMP R2,#100	: WRD.CNT,*	
19491	070056	003743					BLE 6\$		
19492	070060	000437					BR 12\$:	8783
19493	070062	005002				8\$:	CLR R2	: WRD.CNT	8804
19494	070064	010201				9\$:	MOV R2,R1	: WRD.CNT,*	8807
19495	070066	006301					ASL R1		
19496	070070	005761	011600				TST IO.BUF(R1)		
19497	070074	001425					BEQ 11\$		
19498	070076	104455					TRAP 55	:	8810
19499	070100	000140					.WORD 140		
19500	070102	010526					.WORD SYNC		
19501	070104	024052					.WORD DUMPER		
19502	070106	012746	006406				MOV #WRD.10,-(SP)	:	8811
19503	070112	012746	006506				MOV #WRD.19,-(SP)		
19504	070116	012746	010112				MOV #FNC.13,-(SP)		
19505	070122	012746	007552				MOV #PHR.9,-(SP)		
19506	070126	012746	006046				MOV #FOR.FMT,-(SP)		
19507	070132	012746	000005				MOV #5,-(SP)		
19508	070136	010600					MOV SP,R0	: SP,*	
19509	070140	104414					TRAP 14		
19510	070142	062706	000014			10\$:	ADD #14,SP	:	8812
19511	070146	000404					BR 12\$		
19512	070150	005202				11\$:	INC R2	: WRD.CNT	8804
19513	070152	020227	000100				CMP R2,#100	: WRD.CNT,*	
19514	070156	003742					BLE 9\$		
19515	070160	032777	100000	122766		12\$:	BIT #100000,2ML.REG	:	8819
19516	070166	001026					BNE 13\$		
19517	070170	104455					TRAP 55	:	8822
19518	070172	000150					.WORD 150		
19519	070174	010526					.WORD SYNC		
19520	070176	024052					.WORD DUMPER		
19521	070200	012746	010112				MOV #FNC.13,-(SP)	:	8823

```

19523      ;ML4
19524      :
19525
19526 070204 012746 006506      MOV      #WORD.9,-(SP)
19527 070210 012746 006416      MOV      #WORD.11,-(SP)
19528 070214 012746 007366      MOV      #PHR.1,-(SP)
19529 070220 012746 007152      MOV      #WORD.59,-(SP)
19530 070224 012746 006062      MOV      #FIV.FMT,-(SP)
19531 070230 012746 000006      MOV      #6,-(SP)
19532 070234 010600              MOV      SP,R0                ; SP,*
19533 070236 104414              TRAP     14
19534 070240 062706 000016      ADD      #16,SP                ;
19535 070244 104467      13$:    TRAP     67                ;
19536 070246 006000              ROR      R0
19537 070250 103002              BHIS    15$
19538 070252 000167 177346      14$:    JMP      2$
19539 070256 005203      15$:    INC      R3                ; TWICE
19540 070260 020327 000001      CMP      R3,#1                ; TWICE,*
19541 070264 003772              BLE     14$
19542 070266 000207              RTS      PC                    ;
19543
19544      ; Routine Size: 175 words
19545      ; Maximum stack depth per invocation: 11 words
19550
19551
19555
19559 070270      T42::
19560 070270 004767 177236      1$:    JSR      PC,$T42          ;
19561 070274 104466              TRAP     66
19562 070276 006000              ROR      R0
19563 070300 103773              BLO     1$
19564 070302 000207              RTS      PC
19565
19566      ; Routine Size: 6 words
19567      ; Maximum stack depth per invocation: 0 words
19572
19573
19574 :      8830 !<BLF/PAGE>

```


22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (93)

19576 :ML4
19577 :
19578 :
19579 :
19580 :
19581 :
19582 :
19583 :
19584 :
19585 :
19586 :
19587 :
19588 :
19589 :
19590 :
19591 :
19592 :
19593 :
19594 :
19595 :
19596 :
19597 :
19598 :
19599 :
19600 :
19601 :
19602 :
19603 :
19604 :
19605 :
19606 :
19607 :
19608 :
19609 :
19610 :
19611 :
19612 :
19613 :
19614 :
19615 :
19616 :
19617 :
19618 :
19619 :
19620 :
19621 :
19622 :
19623 :
19624 :
19625 :
19626 :
19627 :
19628 :
19629 :
19630 :

8831
8832
8833
8834
8835
8836
8837
8838
8839
8840
8841
8842
8843
8844
8845
8846
8847
8848
8849
8850
8851
8852
8853
8854
8855
8856
8857
8858
8859
8860
8861
8862
8863
8864
8865
8866
8867
8868
8869
8870
8871
8872
8873
8874
8875
8876
8877
8878
8879
8880
8881
8882

BGNTST;

!++

TEST NUMBER: TST 43

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.

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;

!DISABLE ECC

MLWC = not 255;

!LOAD WORD COUNT

MLBA = IO BUF;

!LOAD MBUS ADRS

MLDA = .DSA_CNT;

!LOAD DSA WITH DSA_CNT

MLCS1 = write;

!DO A WRITE FUNCTION

TIME_OUT_LOOP;

if .LBT IS_SET

!SEE IF THE LAST BLOCK XFERR BIT SET

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (93)

```

19632 :ML4
19633 :
19634 :
19635 :      8883      then
19636 :      8884      begin
19637 :      8885      ERRDF (97, ASYNC, DUMPER);
19638 :      8886      PRINTB (THR_FMT, WRD_27, PHR_5, WRD_29);
19639 :      8887      PRINTB (FMT_7, .DSA_CNT);
19640 :      8888      DODU_FLG = ONE;
19641 :      8889      end;
19642 :      8890
19643 :      8891      if .MLDA neq .DSA_CNT + 1
19644 :      8892      then
19645 :      8893      begin
19646 :      8894      ERRDF (98, ASYNC, DUMPER);
19647 :      8895      PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
19648 :      8896      PRINTB (FMT_7, .DSA_CNT);
19649 :      8897      DODU_FLG = ONE;
19650 :      8898      end;
19651 :      8899
19652 :      8900      ENDSUB;
19653 :      8901      end;
19654 :      8902
19655 :      8903      BGNSUB;
19656 :      8904      CLR_MBUS;
19657 :      8905      ECC_DIS = ONE;
19658 :      8906      LAST_BLK_XFER ();
19659 :      8907      MLCST = write;
19660 :      8908      TIME_OUT_LOOP;
19661 :      8909
19662 :      8910      if .MLDA neq .LST_BLK + 1
19663 :      8911      then
19664 :      8912      begin
19665 :      8913      ERRDF (101, ASYNC, DUMPER);
19666 :      8914      PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
19667 :      8915      PRINTB (FMT_7, .LST_BLK);
19668 :      8916      DODU_FLG = ONE;
19669 :      8917      end;
19670 :      8918
19671 :      8919      if .LBT IS_SET
19672 :      8920      then
19673 :      8921      begin
19674 :      8922      MLDA = ONES;
19675 :      8923
19676 :      8924      if .LBT IS_SET
19677 :      8925      then
19678 :      8926      begin
19679 :      8927      ERRDF (99, ASYNC, DUMPER);
19680 :      8928      PRINTB (FIV_FMT, WRD_27, PHR_2, WRD_11, WRD_17, REG_6);
19681 :      8929      DODU_FLG = ONE;
19682 :      8930      end;
19683 :      8931
19684 :      8932      end
19685 :      8933      else
19686 :      8934      begin

```

!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

!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

!ERROR IF NOT

!LBT NOT SET
!ERROR AND SET DODU_FLG

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (93)

19688 :ML4

19689 :

19690 :

19691 :

19692 :

19693 :

19694 :

19695 :

19696 :

19697 :

19698 :

19699 :

19700 :

19701 :

19702 :

19703 :

19704 :

19705 :

19709 :

```

8935     ERRDF (100, ASYNC, DUMPER);
8936     PRINTB (FOR_FMT, WRD_27, PHR_1, WRD_11, WRD_27);
8937     DODU_FLG = ONE;
8938     end;
8939
8940     ENDSUB;
8941
8942     if .DODU_FLG IS_SET
8943     then
8944     begin
8945         DODU (.ML_LUN);
8946         DOCLN;
8947     end;
8948
8949     ENDTST;
    
```

!DROP THIS UNIT IF DODU_FLG SET

19713	070304	004167	113566	\$T43:	JSR	R1,\$SAVE4	:	8829
19714	070310	005001			CLR	R1	: DODU.FLG	8869
19715	070312	016704	121254		MOV	LST.BLK,R4	:	8871
19716	070316	005002			CLR	R2	: DSA.CNT	
19717	070320	000541			BR	6\$		
19718	070322	010203		1\$:	MOV	R2,R3	: DSA.CNT,*	8891
19719	070324	005203			INC	R3		
19720	070326	104402		2\$:	TRAP	2	:	8872
19721	070330	152777	000040	122655	BISB	#40,@ML.REG+40	:	8873
19722	070336	016700	123240		MOV	ML_DUT,R0		
19723	070342	042700	177770		BIC	#177770,R0		
19724	070346	142777	000007	122640	BICB	#7,@ML.REG+40		
19725	070354	150077	122634		BISB	R0,@ML.REG+40		
19726	070360	152777	000002	122704	BISB	#2,@ML.REG+120	:	8875
19727	070366	012777	177400	122570	MOV	#-400,@ML.REG+10	:	8876
19728	070374	012777	011600	122572	MOV	#10.BUF,@ML.REG+20	:	8877
19729	070402	010277	122576		MOV	R2,@ML.REG+30	: DSA.CNT,*	8878
19730	070406	012777	000061	122540	MOV	#61,@ML.REG	:	8879
19731	070414	105777	122604	3\$:	TSTB	@ML.REG+50		
19732	070420	100375			BPL	3\$		
19733	070422	032777	002000	122574	BIT	#2000,@ML.REG+50	:	8882
19734	070430	001433			BEQ	4\$:	
19735	070432	104455			TRAP	5\$:	8885
19736	070434	000141			.WORD	141		
19737	070436	010464			.WORD	ASYNC		
19738	070440	024052			.WORD	DUMPER		
19739	070442	012746	006610		MOV	#WRD.29,-(SP)	:	8886
19740	070446	012746	007472		MOV	#PHR.5,-(SP)		
19741	070452	012746	006602		MOV	#WRD.27,-(SP)		

						22-Dec-1980 09:24:31	TOPS
						22-Dec-1980 09:21:22	PA:<
19743							
19744							
19745							
19746	070456	012746	006034		MOV #THR.FMT,-(SP)		
19747	070462	012746	000004		MOV #4,-(SP)		
19748	070466	010600			MOV SP,R0	: SP,*	
19749	070470	104414			TRAP 14		
19750	070472	010216			MOV R2,(SP)	: DSA.CNT,*	8887
19751	070474	012746	004562		MOV #FMT.7,-(SP)		
19752	070500	012746	000002		MOV #2,-(SP)		
19753	070504	010600			MOV SP,R0	: SP,*	
19754	070506	104414			TRAP 14		
19755	070510	012701	000001		MOV #1,R1	: *,DODU.FLG	8888
19756	070514	062706	000016		ADD #16,SP	:	8884
19757	070520	027703	122460	4\$:	CMP @ML.REG+30,R3	:	8891
19758	070524	001433			BEQ 5\$:	
19759	070526	104455			TRAP 55	:	8894
19760	070530	000142			.WORD 142		
19761	070532	010464			.WORD ASYNC		
19762	070534	024052			.WORD DUMPER		
19763	070536	012746	006442		MOV #WRD.14,-(SP)	:	8895
19764	070542	012746	006632		MOV #WRD.31,-(SP)		
19765	070546	012746	010316		MOV #REG.6,-(SP)		
19766	070552	012746	006034		MOV #THR.FMT,-(SP)		
19767	070556	012746	000004		MOV #4,-(SP)		
19768	070562	010600			MOV SP,R0	: SP,*	
19769	070564	104414			TRAP 14		
19770	070566	010216			MOV R2,(SP)	: DSA.CNT,*	8896
19771	070570	012746	004562		MOV #FMT.7,-(SP)		
19772	070574	012746	000002		MOV #2,-(SP)		
19773	070600	010600			MOV SP,R0	: SP,*	
19774	070602	104414			TRAP 14		
19775	070604	012701	000001		MOV #1,R1	: *,DODU.FLG	8897
19776	070610	062706	000016		ADD #16,SP	:	8893
19777	070614	104467		5\$:	TRAP 67	:	8898
19778	070616	006000			ROR R0		
19779	070620	103642			BLO 2\$		
19780	070622	005202			INC R2	: DSA.CNT	8871
19781	070624	020204		6\$:	CMP R2,R4	: DSA.CNT,*	
19782	070626	002635			BLT 1\$		
19783	070630	104402		7\$:	TRAP 2	:	8901
19784	070632	152777	000040	122354	BISB #40,@ML.REG+40	:	8903
19785	070640	016704	122736		MOV ML.DUT,R4		
19786	070644	042704	177770		BIC #177770,R4		
19787	070650	142777	000007	122336	BICB #7,@ML.REG+40		
19788	070656	150477	122332		BISB R4,@ML.REG+40		
19789	070662	152777	000002	122404	BISB #2,@ML.REG+120	:	8905
19790	070670	004767	125220		JSR PC, LAST.BLK.XFER	:	8906
19791	070674	012777	000061	122252	MOV #61,@ML.REG	:	8907
19792	070702	105777	122316	8\$:	TSTB @ML.REG+50	:	
19793	070706	100375			BPL 8\$		
19794	070710	016702	120656		MOV LST.BLK,R2	:	8910
19795	070714	005202			INC R2		
19796	070716	027702	122262		CMP @ML.REG+30,R2		
19797	070722	001434			BEQ 9\$		

Address	Hex	Hex	Hex	Hex	Label	Comment	Time	Page
19799							22-Dec-1980 09:24:31	TOPS
19800							22-Dec-1980 09:21:22	PA:<
19801								
19802	070724	104455			TRAP	55		8913
19803	070726	000145			.WORD	145		
19804	070730	010464			.WORD	ASYNC		
19805	070732	024052			.WORD	DUMPER		
19806	070734	012746	006442		MOV	#WORD.14,-(SP)		8914
19807	070740	012746	006632		MOV	#WORD.31,-(SP)		
19808	070744	012746	010316		MOV	#REG.6,-(SP)		
19809	070750	012746	006034		MOV	#THR.FMT,-(SP)		
19810	070754	012746	000004		MOV	#4,-(SP)		
19811	070760	010600			MOV	SP,R0	: SP,*	
19812	070762	104414			TRAP	14		
19813	070764	016716	120602		MOV	LST.BLK,(SP)		8915
19814	070770	012746	004562		MOV	#FMT.7,-(SP)		
19815	070774	012746	000002		MOV	#2,-(SP)		
19816	071000	010600			MOV	SP,R0	: SP,*	
19817	071002	104414			TRAP	14		
19818	071004	012701	000001		MOV	#1,R1	: *,DODU.FLG	8916
19819	071010	062706	000016		ADD	#16,SP		8912
19820	071014	032777	002000	122202	BIT	#2000,@ML.REG+50		8919
19821	071022	001440			BEQ	10\$		
19822	071024	012777	177777	122152	MOV	#-1,@ML.REG+30		8922
19823	071032	032777	002000	122164	BIT	#2000,@ML.REG+50		8924
19824	071040	001457			BEQ	11\$		
19825	071042	104455			TRAP	55		8927
19826	071044	000143			.WORD	143		
19827	071046	010464			.WORD	ASYNC		
19828	071050	024052			.WORD	DUMPER		
19829	071052	012746	010316		MOV	#REG.6,-(SP)		8928
19830	071056	012746	006470		MOV	#WORD.17,-(SP)		
19831	071062	012746	006416		MOV	#WORD.11,-(SP)		
19832	071066	012746	007404		MOV	#PHR.2,-(SP)		
19833	071072	012746	006602		MOV	#WORD.27,-(SP)		
19834	071076	012746	006062		MOV	#FIV.FMT,-(SP)		
19835	071102	012746	000006		MOV	#6,-(SP)		
19836	071106	010600			MOV	SP,R0	: SP,*	
19837	071110	104414			TRAP	14		
19838	071112	012701	000001		MOV	#1,R1	: *,DODU.FLG	8929
19839	071116	062706	000016		ADD	#16,SP		8926
19840	071122	000426			BR	11\$		8919
19841	071124	104455			TRAP	55		8935
19842	071126	000144			.WORD	144		
19843	071130	010464			.WORD	ASYNC		
19844	071132	024052			.WORD	DUMPER		
19845	071134	012746	006602		MOV	#WORD.27,-(SP)		8936
19846	071140	012746	006416		MOV	#WORD.11,-(SP)		
19847	071144	012746	007366		MOV	#PHR.1,-(SP)		
19848	071150	012746	006602		MOV	#WORD.27,-(SP)		
19849	071154	012746	006046		MOV	#FOR.FMT,-(SP)		
19850	071160	012746	000005		MOV	#5,-(SP)		
19851	071164	010600			MOV	SP,R0	: SP,*	
19852	071166	104414			TRAP	14		
19853	071170	012701	000001		MOV	#1,R1	: *,DODU.FLG	8937

:ML4
:

122202 9\$:

122152
122164

10\$:

```
19855      ;ML4
19856      ;
19857      ;
19858 071174 062706 000014
19859 071200 104467
19860 071202 006000
19861 071204 103611
19862 071206 005301
19863 071210 001004
19864 071212 016700 122362
19865 071216 104451
19866 071220 104444
19867 071222 000207
19868
19869      ; Routine Size: 232 words
19870      ; Maximum stack depth per invocation: 12 words
19875
19876
19880
19884 071224
19885 071224 004767 177054
19886 071230 104466
19887 071232 006000
19888 071234 103773
19889 071236 000207
19890
19891      ; Routine Size: 6 words
19892      ; Maximum stack depth per invocation: 0 words
19897
19898
19899 :      8950 !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

8934
8938

8942
8945

8829

8947

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (94)

```

19901 :ML4
19902 :
19903 :
19904 :      8951  |
19905 :      8952  |
19906 :      8953  | BGNTST;
19907 :      8954  |
19908 :      8955  | ++
19909 :      8956  | TEST NUMBER:  TST 44
19910 :      8957  |
19911 :      8958  | TEST NAME:    INVALID ADRS TEST
19912 :      8959  |
19913 :      8960  | TEST DESCRIPTION:
19914 :      8961  |
19915 :      8962  |         TEST THE DETECTION OF ILLEGAL DSA
19916 :      8963  |         ADDRESSES BY:
19917 :      8964  |
19918 :      8965  |         1.  DOING A MBUS WRITE FUNCTION
19919 :      8966  |             AT ALL POSSIBLE ILLEGAL DSA
19920 :      8967  |             ADDRESSES AND TEST THE
19921 :      8968  |             IAE BIT SET.
19922 :      8969  |
19923 :      8970  | IMPLICIT INPUTS:
19924 :      8971  |
19925 :      8972  |         IO_BUF
19926 :      8973  |         A VECTOR OF 256 WORDS WHERE
19927 :      8974  |         DATA FOR MBUS READS AND WRITE
19928 :      8975  |         FUNCTION ARE FOUND.
19929 :      8976  |
19930 :      8977  |
19931 :      8978  | ---
19932 :      8979  |
19933 :      8980  | local
19934 :      8981  |     IAE_CNT;
19935 :      8982  |
19936 :      8983  | IAE_CNT = .LST_BLK;
19937 :      8984  |
19938 :      8985  | do
19939 :      8986  |     begin
19940 :      8987  |         IAE_CNT = .IAE_CNT + 1;
19941 :      8988  |         BGNSUB;
19942 :      8989  |         CLR MBUS;
19943 :      8990  |         MLDA = .IAE_CNT;
19944 :      8991  |         MLWC = not 255;
19945 :      8992  |         MLBA = IO_BUF;
19946 :      8993  |         MLCS1 = write;
19947 :      8994  |
19948 :      8995  |         if .IAE IS_NOT_SET
19949 :      8996  |             then
19950 :      8997  |                 begin
19951 :      8998  |                     ERRDF (102, ASYNC, DUMPER);
19952 :      8999  |                     PRINTB (FIV_FMT, WRD_30, PHR_1, WRD_11, WRD_30, WRD_10);
19953 :      9000  |                 end;
19954 :      9001  |
19955 :      9002  |     ENDSUB;

```

```

!INVALID ADRS COUNT
!START AT LAST BLOCK + 1
!TEST FOR ALL INVALID ADDRESSES
!INCREMENT IAE_CNT
!LOAD DSA
!LOAD WORD COUNT
!LOAD MBUS ADRS
!DO A WRITE FUNCTION
!SET IF IAE SET
!ERROR IF NOT SET

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (94)

```

19957 :ML4
19958 :
19959 :
19960 :          9003      end
19961 :          9004      until .IAE_CNT eql %o'177777';
19962 :          9005
19963 :          9006      ENDTST;
19967 :

```

!REPEAT UNTIL ALL TESTED

```

19971 071240 010146          $T44:  MOV      R1,-(SP)          ;
19972 071242 016701 120324      MOV      LST.BLK,R1          ; *,IAE.CNT
19973 071246 005201          1$:      INC      R1          ; IAE.CNT
19974 071250 104402          2$:      TRAP     2          ;
19975 071252 152777 000040 121734      BISB     #40,@ML.REG+40      ;
19976 071260 016700 122316      MOV      ML.DUT,R0
19977 071264 042700 177770      BIC      #177770,R0
19978 071270 142777 000007 121716      BICB     #7,@ML.REG+40
19979 071276 150077 121712      BISB     R0,@ML.REG+40
19980 071302 010177 121676      MOV      R1,@ML.REG+30      ; IAE.CNT,*
19981 071306 012777 177400 121650      MOV      #-400,@ML.REG+10
19982 071314 012777 011600 121652      MOV      #10.BUF,@ML.REG+20
19983 071322 012777 000061 121624      MOV      #61,@ML.REG
19984 071330 032777 002000 121676      BIT      #2000,@ML.REG+60
19985 071336 001026          BNE      3$
19986 071340 104455          TRAP     55
19987 071342 000146          .WORD   146
19988 071344 010464          .WORD   ASYNC
19989 071346 024052          .WORD   DUMPER
19990 071350 012746 006406      MOV      #WORD.10,-(SP)      ;
19991 071354 012746 006624      MOV      #WORD.30,-(SP)
19992 071360 012746 006416      MOV      #WORD.11,-(SP)
19993 071364 012746 007366      MOV      #PHR.1,-(SP)
19994 071370 012746 006624      MOV      #WORD.30,-(SP)
19995 071374 012746 006062      MOV      #FIV.FMT,-(SP)
19996 071400 012746 000006      MOV      #6,-(SP)
19997 071404 010600          MOV      SP,R0          ; SP,*
19998 071406 104414          TRAP     14
19999 071410 062706 000016      ADD      #16,SP
20000 071414 104467          3$:      TRAP     67
20001 071416 006000          ROR      R0
20002 071420 103713          BLO      2$
20003 071422 020127 177777      CMP      R1,#-1          ; IAE.CNT,*
20004 071426 001307          BNE      1$
20005 071430 012601          MOV      (SP)+,R1
20006 071432 000207          RTS      PC

```

: Routine Size: 62 words
: Maximum stack depth per invocation: 8 words

20007
20008
20009

20018
20019
20023
20027 071434
20028 071434 004767 177600
20029 071440 104466
20030 071442 006000
20031 071444 103773
20032 071446 000207

T44::
1\$: JSR PC,\$T44
TRAP 66
ROR R0
BLO 1\$
RTS PC

9004

20033
20034 ; Routine Size: 6 words
20035 ; Maximum stack depth per invocation: 0 words
20040
20041
20042 : 9007 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (95)

```

20044 :ML4
20045 :
20046 :
20047 : 9008 !
20048 : 9009 ! BGNTST;
20049 : 9010 !
20050 : 9011 ! ++
20051 : 9012 ! TEST NUMBER: TST 45
20052 : 9013 !
20053 : 9014 ! TEST NAME: ADRS OVERFLOW BIT TEST
20054 : 9015 !
20055 : 9016 ! TEST DESCRIPTION:
20056 : 9017 ! TEST THE DETECTION OF ADDRESS OVERFLOWS BY:
20057 : 9018 !
20058 : 9019 ! 1. STARTING AT THE LAST BLOCK DO A TWO BLOCK TRANSFER.
20059 : 9020 !
20060 : 9021 ! 2. READ THE AOE BIT SET.
20061 : 9022 !
20062 : 9023 ! IMPLICIT INPUTS:
20063 : 9024 ! IO_BUF
20064 : 9025 ! A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
20065 : 9026 ! FUNCTIONS ARE FOUND.
20066 : 9027 !
20067 : 9028 !
20068 : 9029 ! CLR MBUS;
20069 : 9030 ! MLWC = not 511; !DO TWO BLOCK XFFRR
20070 : 9031 ! MLBA = IO_BUF; !LOAD UBUS ADRS
20071 : 9032 ! MLDA = .LST_BLK; !START AT LAST BLOCK
20072 : 9033 ! MLCS1 = write; !DO A WRITE FUNCTION
20073 : 9034 ! TIME_OUT_LOOP;
20074 : 9035 !
20075 : 9036 ! if .AOE IS_NOT_SET !SEE IF AOE SET
20076 : 9037 ! then
20077 : 9038 ! begin !ERROR IF NOT SET
20078 : 9039 ! ERRDF (103, SYNC, DUMPER);
20079 : 9040 ! PRINTB (FOR_FMT, WRD_26, PHR_1, WRD_11, FNC_19);
20080 : 9041 ! end;
20081 : 9042 !
20082 : 9043 ! ENDTST;
20086 :
20090 071450 152777 000040 121536 $T45: B1SB #40,@ML.REG+40 ; 9009
20091 071456 016700 122120 MOV ML.DUT,RO
20092 071462 042700 177770 BIC #177770,RO
20093 071466 142777 000007 121520 BICB #7,@ML.REG+40
20094 071474 150077 121514 B1SB RO,@ML.REG+40
20095 071500 012777 177000 121456 MOV #-1000,@ML.REG+10 ; 9030
20096 071506 012777 011600 121460 MOV #IO.BUF,@ML.REG+20 ; 9031
20097 071514 016777 120052 121462 MOV LST.BLK,@ML.REG+30 ; 9032

```

```

20099          ;ML4
20100          ;
20101
20102 071522 012777 000061 121424      MOV    #61,@ML.REG      ;
20103 071530 105777 121470      1$:   TSTB   @ML.REG+50
20104 071534 100375                BPL    1$
20105 071536 032777 001000 121470      BIT    #1000,@ML.REG+60 ;
20106 071544 001024                BNE    2$
20107 071546 104455                TRAP   55                ;
20108 071550 000147                .WORD 147
20109 071552 010526                .WORD SYNC
20110 071554 024052                .WORD DUMPER
20111 071556 012746 010204      MOV    #FNC.19,-(SP)    ;
20112 071562 012746 006416      MOV    #WRD.11,-(SP)
20113 071566 012746 007366      MOV    #PHR.1,-(SP)
20114 071572 012746 006574      MOV    #WRD.26,-(SP)
20115 071576 012746 006046      MOV    #FOR.FMT,-(SP)
20116 071602 012746 000005      MOV    #5,-(SP)
20117 071606 010600                MOV    SP,R0           ; SP,*
20118 071610 104414                TRAP   14
20119 071612 062706 000014      ADD    #14,SP
20120 071616 000207      2$:   RTS    PC
20121
20122          ; Routine Size: 52 words
20123          ; Maximum stack depth per invocation: 6 words
20128
20129
20133
20137 071620      T45::
20138 071620 004767 177624      1$:   JSR    PC,$T45
20139 071624 104466                TRAP   66
20140 071626 006000                ROR    R0
20141 071630 103773                BLO    1$
20142 071632 000207                RTS    PC
20143
20144          ; Routine Size: 6 words
20145          ; Maximum stack depth per invocation: 0 words
20150
20151
20152 ;          9044 !<BLF/PAGE>

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

9033

9036

9039

9040

9038

9006

9041

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (96)

```

20154 :ML4
20155 :
20156 :
20157 :          9045 !
20158 :          9046 BGNTST:
20159 :          9047
20160 :          9048 !++
20161 :          9049 TEST NUMBER: TST 46
20162 :          9050
20163 :          9051 TEST NAME: SYNC BUS PARITY TEST
20164 :          9052
20165 :          9053 TEST DESCRIPTION:
20166 :          9054 TEST ABILITY OF SYNC BUS TO DETECT AND GENERATE
20167 :          9055 GOOD PARITY BY:
20168 :          9056
20169 :          9057 1. VIA MBUS WRITE FUNCTION WRITE ALTERNATING ONES AND ZEROES TO THE
20170 :          9058 DEVICE AND READ THE DPAR BIT CLEARED.
20171 :          9059
20172 :          9060 2. REPEAT WITH SHIFTED DATA
20173 :          9061
20174 :          9062 3. VIA MBUS READ FUNCTION READ THE ALTERNATING PATTERN AND MDPE CLEARED.
20175 :          9063
20176 :          9064 IMPLICIT INPUTS:
20177 :          9065 IO_BUF
20178 :          9066 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITES
20179 :          9067 FUNCTIONS ARE FOUND.
20180 :          9068
20181 :          9069
20182 :          9070 if .PAR_DIS IS_SET !SEE IF PARITY IS DISABLED
20183 :          9071 then !PRINT MESSAGE AND EXIT TST IF YES
20184 :          9072 begin
20185 :          9073 PRINTB (THR_FMT, FNC_3, WRD_7, WRD_37);
20186 :          9074 EXIT_TST;
20187 :          9075 end;
20188 :          9076
20189 :          9077 CLR_MBUS;
20190 :          9078 BAI = ONE; !SET ON FIRST IO BUF ADRS
20191 :          9079 IO_BUF = %o'125252'; !ALTERNATE 1, 0 PATTERN
20192 :          9080
20193 :          9081 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
20194 :          9082 begin
20195 :          9083 BGNSUB;
20196 :          9084 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
20197 :          9085 MLCS1 = write; !DO A WRITE FUNCTION
20198 :          9086 TIME_OUT_LOOP;
20199 :          9087
20200 :          9088 if .DPAR IS_SET !SEE IF DPAR GOT SET
20201 :          9089 then !ERROR IF SET
20202 :          9090 begin
20203 :          9091 ERRDF (105, SYNC, DUMPER);
20204 :          9092 PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_9);
20205 :          9093 end;
20206 :          9094
20207 :          9095 IO_BUF = .IO_BUF^ONE; !SHIFT THE IO BUF & REPEAT
20208 :          9096 MLER = ZEROES; !CLEAR OUT ERROR REG & REPEAT

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (96)

```

20210 ;ML4
20211 :
20212 :
20213 :          9097      ENDSUB;
20214 :          9098      end;
20215 :          9099
20216 :          9100      incr TWICE from 0 to 1 do          !REPEATE LOOP TWICE
20217 :          9101      begin
20218 :          9102      BGN SUB;
20219 :          9103      CLR_MBUS;
20220 :          9104      BAI = ONE;          !SET ON FIRST IO BUF ADRS
20221 :          9105      GD_BLK_XFER ();          !SET UP A GOOD BLOCK XFERR
20222 :          9106      MLCS1 = read;          !DO A READ XFERR
20223 :          9107      TIME_OUT_LOOP;
20224 :          9108
20225 :          9109      if .MDPE IS_SET          !SEE IF READ GENERATED BAD PARITY
20226 :          9110      then
20227 :          9111      begin          !ERROR IF MDPE SET
20228 :          9112      ERRDF (106, SYNC, DUMPER);
20229 :          9113      PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_8);
20230 :          9114      end;
20231 :          9115
20232 :          9116      ENDSUB;
20233 :          9117      IO_BUF = .IO_BUF^ONE;          !MAKE DATA PATTERN HAVE ONE LESS ONE
20234 :          9118      GD_BLK_XFER ();          !SET UP A GOOD BLK XFER
20235 :          9119      MLCS1 = write;          !CHANGE THE BACKGROUND IN MEMORY
20236 :          9120      TIME_OUT_LOOP;
20237 :          9121      end;
20238 :          9122
20239 :          9123      ENDTST;

```

```

20247 071634 010146          $T46:  MOV    R1, -(SP)          ;          9043
20248 071636 026727 117724 000001  CMP    PAR.DIS, #1          ;          9070
20249 071644 001021          BNE    1$
20250 071646 012746 006700          MOV    #WRD.37, -(SP)          ;          9073
20251 071652 012746 006344          MOV    #WRD.7, -(SP)
20252 071656 012746 007750          MOV    #FNC.3, -(SP)
20253 071662 012746 006034          MOV    #THR.FMT, -(SP)
20254 071666 012746 000004          MOV    #4, -(SP)
20255 071672 010600          MOV    SP, R0          ; SP,*
20256 071674 104414          TRAP  14
20257 071676 104463          TRAP  63
20258 071700 062706 000012          ADD    #12, SP          ;          9070
20259 071704 000167 000402          JMP    9$          ;          9072
20260 071710 152777 000040 121276 1$:  BISB  #40, @ML.REG+40          ;          9075
20261 071716 016701 121660          MOV    ML.DUT, R1
20262 071722 042701 177770          BIC    #177770, R1
20263 071726 142777 000007 121260          BICB  #7, @ML.REG+40

```

20265													22-Dec-1980 09:24:31	TOPS
20266													22-Dec-1980 09:21:22	PA: <
20267														
20268	071734	150177	121254			BISB	R1, @ML.REG+40							
20269	071740	152777	000010	121246		BISB	#10, @ML.REG+40							9078
20270	071746	012767	125252	117624		MOV	#-52526, IO.BUF							9079
20271	071754	005001				CLR	R1							9081
20272	071756	104402			2\$:	TRAP	2							9082
20273	071760	004767	124076			JSR	PC, GD.BLK.XFER							9084
20274	071764	012777	000061	121162		MOV	#61, @ML.REG							9085
20275	071772	105777	121226		3\$:	TSTB	@ML.REG+50							
20276	071776	100375				BPL	3\$							
20277	072000	132777	000040	121226		BITB	#40, @ML.REG+60							9088
20278	072006	001424				BEQ	4\$							
20279	072010	104455				TRAP	5\$							9091
20280	072012	000151				.WORD	151							
20281	072014	010526				.WORD	SYNC							
20282	072016	024052				.WORD	DUMPER							
20283	072020	012746	006374			MOV	#WRD.9, -(SP)							9092
20284	072024	012746	006344			MOV	#WRD.7, -(SP)							
20285	072030	012746	006336			MOV	#WRD.6, -(SP)							
20286	072034	012746	006544			MOV	#WRD.23, -(SP)							
20287	072040	012746	006046			MOV	#FOR.FMT, -(SP)							
20288	072044	012746	000005			MOV	#5, -(SP)							
20289	072050	010600				MOV	SP, R0							
20290	072052	104414				TRAP	14							
20291	072054	062706	000014			ADD	#14, SP							9090
20292	072060	006367	117514		4\$:	ASL	IO.BUF							9095
20293	072064	005077	121144			CLR	@ML.REG+60							9096
20294	072070	104467				TRAP	67							
20295	072072	006000				ROR	R0							
20296	072074	103730				BLO	2\$							
20297	072076	005201				INC	R1							
20298	072100	020127	000001			CMP	R1, #1							9081
20299	072104	003724				BLE	2\$							
20300	072106	005001				CLR	R1							
20301	072110	104402			5\$:	TRAP	2							9100
20302	072112	152777	000040	121074		BISB	#40, @ML.REG+40							9101
20303	072120	016700	121456			MOV	ML, DUT, R0							9102
20304	072124	042700	177770			BIC	#177770, R0							
20305	072130	142777	000007	121056		BICB	#7, @ML.REG+40							
20306	072136	150077	121052			BISB	R0, @ML.REG+40							
20307	072142	152777	000010	121044		BISB	#10, @ML.REG+40							9104
20308	072150	004767	123706			JSR	PC, GD.BLK.XFER							9105
20309	072154	012777	000071	120772		MOV	#71, @ML.REG							9106
20310	072162	105777	121036		6\$:	TSTB	@ML.REG+50							
20311	072166	100375				BPL	6\$							
20312	072170	032777	000400	121016		BIT	#400, @ML.REG+40							9109
20313	072176	001424				BEQ	7\$							
20314	072200	104455				TRAP	55							9112
20315	072202	000152				.WORD	152							
20316	072204	010526				.WORD	SYNC							
20317	072206	024052				.WORD	DUMPER							
20318	072210	012746	006360			MOV	#WRD.8, -(SP)							9113
20319	072214	012746	006344			MOV	#WRD.7, -(SP)							

```
20321      ;ML4
20322      ;
20323
20324 072220 012746 006336      MOV      #WRD.6, -(SP)
20325 072224 012746 006544      MOV      #WRD.23, -(SP)
20326 072230 012746 006046      MOV      #FOR.FMT, -(SP)
20327 072234 012746 000005      MOV      #5, -(SP)
20328 072240 010600      MOV      SP, R0      ; SP, *
20329 072242 104414      TRAP     14
20330 072244 062706 000014      ADD      #14, SP
20331 072250 104467      7$: TRAP     67
20332 072252 006000      ROR      R0
20333 072254 103715      BLO      5$
20334 072256 006367 117316      ASL      IO.BUF
20335 072262 004767 123574      JSR      PC, GD.BLK.XFER
20336 072266 012777 000061 120660      MOV      #61, @ML.REG
20337 072274 105777 120724      8$: TSTB    @ML.REG+50
20338 072300 100375      BPL      8$
20339 072302 005201      INC      R1      ; TWICE
20340 072304 020127 000001      CMP      R1, #1    ; TWICE, *
20341 072310 003677
20342 072312 012601      9$: MOV      (SP)+, R1
20343 072314 000207      RTS      PC
20344
20345      ; Routine Size: 153 words
20346      ; Maximum stack depth per invocation: 7 words
20351
20352
20356
20360 072316      T46::
20361 072316 004767 177312      1$: JSR      PC, $T46
20362 072322 104466      TRAP     66
20363 072324 006000      ROR      R0
20364 072326 103773      BLO      1$
20365 072330 000207      RTS      PC
20366
20367      ; Routine Size: 6 words
20368      ; Maximum stack depth per invocation: 0 words
20373
20374 ;      9124 !<BLF/PAGE>
```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BL:ss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (97)

```

20376 :ML4
20377 :
20378 :
20379 : 9125 :
20380 : 9126 :
20381 : 9127 : BGNTST;
20382 : 9128 :
20383 : 9129 : !++
20384 : 9130 : TEST NUMBER: TST 47
20385 : 9131 :
20386 : 9132 : TEST NAME: WRITE READ ML11
20387 : 9133 :
20388 : 9134 : TEST DESCRIPTION:
20389 : 9135 :
20390 : 9136 : PROVIDES A MBUS READ/WRITE
20391 : 9137 : DATA TRANSFER TROUBLE SHOOTING
20392 : 9138 : LOOP BY:
20393 : 9139 :
20394 : 9140 : 1. LOAD APPROPRIATE RH REGISTERS.
20395 : 9141 : DO A WRITE FUNCTION.
20396 : 9142 :
20397 : 9143 : 2. LOAD APPROPRIATE RH REGISTERS
20398 : 9144 : DO A WRITE CHECK FUNCTION.
20399 : 9145 :
20400 : 9146 : 3. COMPLIMENT DATA AND
20401 : 9147 : REPEAT.
20402 : 9148 :
20403 : 9149 : IMPLICIT INPUTS:
20404 : 9150 :
20405 : 9151 : IO_BUF
20406 : 9152 : A VECTOR OF 256 WORDS WHERE
20407 : 9153 : DATA FOR MBUS READ AND WRITE
20408 : 9154 : FUNCTIONS IS FOUND.
20409 : 9155 :
20410 : 9156 :
20411 : 9157 :
20412 : 9158 : local
20413 : 9159 : TST_PAT; !TEST PATTERN
20414 : 9160 :
20415 : 9161 : TST_PAT = %o'052525';
20416 : 9162 :
20417 : 9163 : incr TWICE from 0 to 1 do !WRITE READ 1'S AND 0'S ON MBUS
20418 : 9164 : begin
20419 : 9165 : CLR_MBUS;
20420 : 9166 : BAI = ONE; !SET ON FIRST IO BUF ADRS
20421 : 9167 : IO_BUF = TST_PAT; !LOAD FIRST IO BUF ADRS
20422 : 9168 : GO_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
20423 : 9169 : MLTST = write; !DO A WRITE FUNCTION
20424 : 9170 : TIME_OUT_LOOP;
20425 : 9171 : CLR_MBUS;
20426 : 9172 : BAI = ONE; !SET ON FIRST IO BUF ADRS
20427 : 9173 : GO_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
20428 : 9174 : MLTST = WRT_CHK; !DO A WRITE CHECK FUNCTION
20429 : 9175 : TIME_OUT_LOOP;
20430 : 9176 :

```


20432 :ML4
20433 :
20434 :
20435 :
20436 :
20437 :
20438 :
20439 :
20440 :
20441 :
20442 :
20443 :
20444 :
20445 :
20449 :

```

9177   if .WCE IS_SET
9178   then
9179   begin
9180     ERRDF (109, TRBLE_LOOP, DUMPER);
9181     PRINTB (SIX_FMT, FNC_4, WRD_10, WRD_12, FNC_5, FNC_6, FNC_3);
9182   end;
9183
9184   TST_PAT = not .TST_PAT;
9185   end;
9186
9187   ENDTST;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (97)

```

!SEE IF WRITE CHECK ERROR SET
!ERROR IF SET
!COMPLIMENT TST_PAT AND REPEAT

```

20453	072332	004167	111506	\$T47:	JSR	R1,\$SAVE2	:	9123	
20454	072336	012701	052525		MOV	#52525,R1	:	9161	
20455	072342	005002			CLR	R2	:	9163	
20456	072344	152777	000040	120642	1\$:	BISB	#40,@ML.REG+40	:	9164
20457	072352	016700	121224		MOV	ML.DUT,R0	:		
20458	072356	042700	177770		BIC	#177770,R0	:		
20459	072362	142777	000007	120624		BICB	#7,@ML.REG+40	:	
20460	072370	150077	120620		BISB	R0,@ML.REG+40	:		
20461	072374	152777	000010	120612		BISB	#10,@ML.REG+40	:	9166
20462	072402	010167	117172		MOV	R1,IO.BUF	:	9167	
20463	072406	004767	123450		JSR	PC,GD.BLK.XFER	:	9168	
20464	072412	012777	000061	120534		MOV	#61,@ML.REG	:	9169
20465	072420	105777	120600	2\$:	TSTB	@ML.REG+50	:		
20466	072424	100375			BPL	2\$:		
20467	072426	152777	000040	120560		BISB	#40,@ML.REG+40	:	9170
20468	072434	016700	121142		MOV	ML.DUT,R0	:		
20469	072440	042700	177770		BIC	#177770,R0	:		
20470	072444	142777	000007	120542		BICB	#7,@ML.REG+40	:	
20471	072452	150077	120536		BISB	R0,@ML.REG+40	:		
20472	072456	152777	000010	120530		BISB	#10,@ML.REG+40	:	9172
20473	072464	004767	123372		JSR	PC,GD.BLK.XFER	:	9173	
20474	072470	012777	000051	120456		MOV	#51,@ML.REG	:	9174
20475	072476	105777	120522	3\$:	TSTB	@ML.REG+50	:		
20476	072502	100375			BPL	3\$:		
20477	072504	032777	040000	120502		BIT	#40000,@ML.REG+40	:	9177
20478	072512	001430			BEQ	4\$:		
20479	072514	104455			TRAP	5\$:	9180	
20480	072516	000155			.WORD	155	:		
20481	072520	011026			.WORD	TRBLE_LOOP	:		
20482	072522	024052			.WORD	DUMPER	:		
20483	072524	012746	007750		MOV	#FNC.3,-(SP)	:	9181	
20484	072530	012746	010004		MOV	#FNC.6,-(SP)	:		
20485	072534	012746	007774		MOV	#FNC.5,-(SP)	:		

20487
 20488
 20489
 20490
 20491
 20492
 20493
 20494
 20495
 20496
 20497
 20498
 20499
 20500
 20501
 20502
 20503
 20504
 20505
 20510
 20511
 20515
 20519
 20520
 20521
 20522
 20523
 20524
 20525
 20526
 20527
 20532
 20533
 20534

072540 012746 006426
 072544 012746 006406
 072550 012746 007756
 072554 012746 006100
 072560 012746 000007
 072564 010600
 072566 104414
 072570 062706 000020
 072574 005101
 072576 005202
 072600 020227 000001
 072604 003657
 072606 000207

:ML4
 :

MOV #WRD.12,-(SP)
 MOV #WRD.10,-(SP)
 MOV #FNC.4,-(SP)
 MOV #SIX.FMT,-(SP)
 MOV #7,-(SP)
 MOV SP,R0
 TRAP 14
 ADD #20,SP
 4\$: COM R1
 INC R2
 CMP R2,#1
 BLE 1\$
 RTS PC

: SP,*
 :
 : TST.PAT
 : TWICE
 : TWICE,*
 :

: Routine Size: 87 words
 : Maximum stack depth per invocation: 11 words

T47::
 1\$:

JSR PC,\$T47
 TRAP 66
 ROR R0
 BLO 1\$
 RTS PC

:

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

9188 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
 22-Dec-1980 09:21:22 PA:<

9179
 9184
 9163
 9123

9185

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (98)

```

20536 :ML4
20537 :
20538 :
20539 : 9189 !
20540 : 9190 ! BGNTST;
20541 : 9191 !
20542 : 9192 ! ++
20543 : 9193 ! TEST NUMBER: TST 48
20544 : 9194 !
20545 : 9195 ! TEST NAME: CRC DATA BUS TEST
20546 : 9196 !
20547 : 9197 ! TEST DESCRIPTION:
20548 : 9198 ! TEST THE CRC DATA BUS BETWEEN THE CRC
20549 : 9199 ! GENERATORS AND THE CRC/MBUS DATA
20550 : 9200 ! MUX FOR CONTINUITY AND BIT UNIQUENESS
20551 : 9201 ! BY:
20552 : 9202 !
20553 : 9203 ! 1. WHILE IN ECC DIAGNOSTIC MODE DO
20554 : 9204 ! GOOD BLOCK TRANSFERS WITH VARIOUS
20555 : 9205 ! CRC A,CRC B AND PAR CRC WRD WHICH
20556 : 9206 ! WILL EXERCISE BUS CONTIRUITY AND
20557 : 9207 ! UNIQUENESS.
20558 : 9208 !
20559 : 9209 ! 2. THEN IN DATA DIAGNOSTEC MODE READ
20560 : 9210 ! THE CRC NIBBLES FROM ONE CRC GROUP
20561 : 9211 ! FOR CORRECT CRC_A,CRC_B AND PAR_CRC_WRD
20562 : 9212 ! DATA PATTERNS.
20563 : 9213 !
20564 : 9214 ! IMPLICIT INPUTS:
20565 : 9215 ! PD TEMP
20566 : 9216 ! A BIT VECTOR OF 16 BITS WHERE THE READ
20567 : 9217 ! PROM DATA IS STORED OAND ACCESSED FROM.
20568 : 9218 !
20569 : 9219 !
20570 : 9220 ! --
20571 : 9221 !
20572 : 9222 ! local
20573 : 9223 ! CRC_NIB,
20574 : 9224 ! CRC_TSTED,
20575 : 9225 ! ERR_FLG,
20576 : 9226 ! NIB_PAT;
20577 : 9227 !
20578 : 9228 ! CRC_NIB = 9;
20579 : 9229 !
20580 : 9230 ! incr LOOP from 0 to 4 do
20581 : 9231 ! begin
20582 : 9232 ! BGNSUB;
20583 : 9233 ! CLR_MBUS;
20584 : 9234 ! ECC_DM = ONE;
20585 : 9235 !
20586 : 9236 ! case .LOOP from 0 to 4 of
20587 : 9237 ! set
20588 : 9238 !
20589 : 9239 ! [0] :
20590 : 9240 ! NIB_PAT = %b'110';

```

```

! POINTER TO THE CRC NIBBLE
! STORES HOW MANY CRC NIBBLES WE HAVE TESTED
! ERROR FLAG
! TEST DATA FOR THE CRC NIBBLE
! SET CRC NIBBLE POINTER
! REPEAT THE TEST WITH FIVE NIBBLE PATTERNS
! START OF THE SCOPE LOOP
! SET ECC DIAG MOD
! SELECT ONE OF THE FIVE NIBBLE PATTERNS
! PATTERN 0

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (98)

```

20592 :ML4
20593 :
20594 :
20595 :          9241
20596 :          9242      [1] :          !PATTERN 1
20597 :          9243      begin
20598 :          9244      MLE1 = ONES;
20599 :          9245      MLE2 = ONES;
20600 :          9246      NIB_PAT = %b'001';
20601 :          9247      end;
20602 :          9248
20603 :          9249      [2] :          !PATTERN 2
20604 :          9250      begin
20605 :          9251      CRC_A = ONES;
20606 :          9252      CRC_B = ONES;
20607 :          9253      NIB_PAT = %b'000';
20608 :          9254      end;
20609 :          9255
20610 :          9256      [3] :          !PATTERN 3
20611 :          9257      begin
20612 :          9258      PAR_CRC_WRD = ONES;
20613 :          9259      CRC_B = ONES;
20614 :          9260      NIB_PAT = %b'011';
20615 :          9261      end;
20616 :          9262
20617 :          9263      [4] :          !PATTERN 4
20618 :          9264      begin
20619 :          9265      PAR_CRC_WRD = ONES;
20620 :          9266      CRC_A = ONES;
20621 :          9267      NIB_PAT = %b'101';
20622 :          9268      end;
20623 :          9269      tes;
20624 :          9270
20625 :          9271      GD_BLK_XFER ();          !SET UP A GOOD BLOCK TRANSFER
20626 :          9272      MLCS1 = write;          !DO A WRITE TRANSFER
20627 :          9273      TIME_OUT_LOOP;          !WAIT UNTIL DRIVE READY
20628 :          9274      DAT_DM = ONE;          !SET DATA DIAG MODE
20629 :          9275      GD_BLK_XFER ();          !SET UP A GOOD BLOCK TRANSFER
20630 :          9276      MLCS1 = read;          !DO A READ TRANSFER
20631 :          9277      CRC_TSTED = ZERO;
20632 :          9278      DELAY (ONE_US);
20633 :          9279
20634 :          9280      do          !TEST ONE CRC GROUP '6 CRC NIBBLES'
20635 :          9281      begin
20636 :          9282      PD_TEMP = .MLPD;          !READ THE PROM DATA FOR THIS ARRAY WORD
20637 :          9283      DAT_CLK = ONE;          !CLOCK OUT AN ARRAY WORD INTO THE DATA DIAG REG'S
20638 :          9284      DELAY (ONE_US);          !DELAY ONE MICRO SECOND
20639 :          9285      RD_LNG_WRD;          !READ THE DATA DIAG REG'S INTO MEMORY
20640 :          9286
20641 :          9287      if .PD_TEMP [.CRC_NIB] IS_NOT_SET          !TEST THIS CRC NIBBLE IF GOOD
20642 :          9288      then
20643 :          9289      begin
20644 :          9290      CRC_TSTED = .CRC_TSTED + 1;          !INCREMENT THE GOOD NIBBLE COUNTER
20645 :          9291      TST_LNG_WRD (.CRC_NIB, .NIB_PAT, ERR_FLG);          !TEST THE CRC NIBBLE
20646 :          9292

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (98)

```

20648 :ML4
20649 :
20650 :
20651 :          9293          if .ERR_FLG IS_SET          !SEE IF THE TEST FOUND AN ERROR
20652 :          9294          then
20653 :          9295          begin          !REPORT ERROR IF FLAG IS SET
20654 :          9296          ERRDF (131, SYNC, DUMPER);
20655 :          9297          PRINTB (FOR_FMT, WRD 64, WRD 24, WRD 73, WRD 10);
20656 :          9298          PRINTB (FMT_5, .NIB_PAT, .NIB_SAVE [NIB_9], 8);
20657 :          9299          end;
20658 :          9300
20659 :          9301          end;
20660 :          9302
20661 :          9303          end
20662 :          9304          until .CRC_TSTED eql 6;          !REPEAT UNTIL ONE CRC GROUP HAS BEEN TESTED
20663 :          9305
20664 :          9306          ENDSUB;          !END OF SCOPE LOOP
20665 :          9307          end;
20666 :          9308
20667 :          9309          ENDTST;
20671 :
20675 072624 004167 111266          $T48: JSR          R1,$SAVE5          ;          9187
20676 072630 162706 000006          SUB          #6,SP
20677 072634 012705 000011          MOV          #11,R5          ; *,CRC.NIB          9228
20678 072640 010546          MOV          R5,-(SP)          ; CRC.NIB,*          9287
20679 072642 006216          ASR          (SP)
20680 072644 006216          ASR          (SP)
20681 072646 006216          ASR          (SP)
20682 072650 062716 013120          ADD          #PD.TEMP,(SP)
20683 072654 005001          CLR          R1          ; LOOP          9230
20684 072656 104402          TRAP         2          ;          9231
20685 072660 152777 000040 120326          BISB         #40,@ML.REG+40          ;          9232
20686 072666 016704 120710          MOV          ML.DUT,R4
20687 072672 042704 177770          BIC          #177770,R4
20688 072676 142777 000007 120310          BICB         #7,@ML.REG+40
20689 072704 150477 120304          BISB         R4,@ML.REG+40
20690 072710 152777 000001 120356          BISB         #1,@ML.REG+120          ;          9234
20691 072716 010104          MOV          R1,R4          ; LOOP,*          9236
20692 072720 006304          ASL          R4
20693 072722 066407 072726          ADD          2$(R4),PC
20694 072726 000012          .WORD        3$-2$          2$:
20695 072730 000020          .WORD        4$-2$
20696 072732 000042          .WORD        5$-2$
20697 072734 000062          .WORD        6$-2$
20698 072736 000104          .WORD        7$-2$
20699 072740 012702 000006          3$: MOV          #6,R2          ; *,NIB.PAT          9240
20700 072744 000442          BR           8$          ;          9236
20701 072746 012777 177777 120350          4$: MOV          #-1,@ML.REG+150          ;          9244

```

Address	Instruction	Comments	Address	Instruction	Comments	Date/Time	Page
20703						22-Dec-1980 09:24:31	TOPS
20704						22-Dec-1980 09:21:22	PA:C
20705							
20706	MOV	#-1,@ML.REG+160	120352				9245
20707	MOV	#1,R2			: *,NIB.PAT		9246
20708	BR	8S			:		9236
20709	BISB	#77,@ML.REG+150	120326	5S:			9251
20710	BISB	#77,@ML.REG+160	120330				9252
20711	CLR	R2			: NIB.PAT		9253
20712	BR	8S			:		9236
20713	BIS	#37400,@ML.REG+150	120306	6S:			9258
20714	BISB	#77,@ML.REG+160	120310				9259
20715	MOV	#3,R2			: *,NIB.PAT		9260
20716	BR	8S			:		9236
20717	BIS	#37400,@ML.REG+150	120264	7S:			9265
20718	BISB	#77,@ML.REG+150	120256				9266
20719	MOV	#5,R2			: *,NIB.PAT		9267
20720	JSR	PC,GD,BLK.XFER		8S:			9271
20721	MOV	#61,@ML.REG	120070		:		9272
20722	TSTB	@ML.REG+50		9S:			
20723	BPL	9S			:		
20724	BISB	#10,@ML.REG+120	120174		:		9274
20725	JSR	PC,GD,BLK.XFER			:		9275
20726	MOV	#71,@ML.REG	120042		:		9276
20727	CLR	2(SP)			: CRC.TSTED		9277
20728	MOV	#1,R3			: *,SSTMP2		9278
20729	BEQ	13S		10S:			
20730	MOV	LSDLY,R4			: *,SSTMP1		
20731	BEQ	12S			:		
20732	CLR	6(SP)		11S:	: SSTMP		
20733	DEC	R4			: SSTMP1		
20734	BNE	11S			:		
20735	DEC	R3		12S:	: SSTMP2		
20736	BR	10S			:		
20737	MOV	@ML.REG+230,PD.TEMP	120232	13S:			9282
20738	BISB	#20,@ML.REG+120	120112		:		9283
20739	MOV	#1,R3			: *,SSTMP2		9284
20740	BEQ	17S		14S:			
20741	MOV	LSDLY,R4			: *,SSTMP1		
20742	BEQ	16S			:		
20743	CLR	6(SP)		15S:	: SSTMP		
20744	DEC	R4			: SSTMP1		
20745	BNE	15S			:		
20746	DEC	R3		16S:	: SSTMP2		
20747	BR	14S			:		
20748	MOV	@ML.REG+170,D1.TEMP	120126	17S:			
20749	MOV	@ML.REG+200,D2.TEMP	120130				
20750	MOV	@ML.REG+160,E2.TEMP	120102				
20751	MOV	(SP),-(SP)			:		9287
20752	MOV	R5,-(SP)			: CRC.NIB, *		
20753	BIC	#177770,(SP)			:		
20754	MOV	#1,-(SP)			:		
20755	CLR	-(SP)			:		
20756	JSR	PC,BL\$GT2			:		
20757	ADD	#10,SP			:		

Address	OpCode	Operand 1	Operand 2	Operand 3	Comments	Line #
20759						
20760						
20761						
20762	073262	005700			TST R0	
20763	073264	001066			BNE 19\$	
20764	073266	005266	000002		INC 2(SP)	: CRC.TSTED 9290
20765	073272	010546			MOV R5, -(SP)	: CRC.NIB,* 9291
20766	073274	010246			MOV R2, -(SP)	: NIB.PAT,*
20767	073276	012746	000012		MOV #12, -(SP)	
20768	073302	060616			ADD SP, (SP)	: ERR.FLG,*
20769	073304	004767	122636		JSR PC, TST.LNG.WRD	
20770	073310	026627	000012	000001	CMP 12(SP), #1	: ERR.FLG,* 9293
20771	073316	001047			BNE 18\$	
20772	073320	104455			TRAP 55	:
20773	073322	000203			.WORD 203	
20774	073324	010526			.WORD SYNC	
20775	073326	024052			.WORD DUMPER	
20776	073330	012746	006406		MOV #WRD.10, -(SP)	:
20777	073334	012746	007310		MOV #WRD.73, -(SP)	
20778	073340	012746	006552		MOV #WRD.24, -(SP)	
20779	073344	012746	007224		MOV #WRD.64, -(SP)	
20780	073350	012746	006046		MOV #FOR.FMT, -(SP)	
20781	073354	012746	000005		MOV #5, -(SP)	
20782	073360	010600			MOV SP, R0	: SP,*
20783	073362	104414			TRAP 14	:
20784	073364	012716	000010		MOV #10, (SP)	
20785	073370	016704	115562		MOV NIB.SAVE+4, R4	: 9298
20786	073374	006204			ASR R4	
20787	073376	006204			ASR R4	
20788	073400	006204			ASR R4	
20789	073402	006204			ASR R4	
20790	073404	000304			SWAB R4	
20791	073406	042704	177770		BIC #177770, R4	
20792	073412	010446			MOV R4, -(SP)	
20793	073414	010246			MOV R2, -(SP)	: NIB.PAT,*
20794	073416	012746	004430		MOV #FMT.5, -(SP)	
20795	073422	012746	000004		MOV #4, -(SP)	
20796	073426	010600			MOV SP, R0	: SP,*
20797	073430	104414			TRAP 14	:
20798	073432	062706	000024		ADD #24, SP	:
20799	073436	062706	000006		ADD #6, SP	:
20800	073442	026627	000002	000006	CMP 2(SP), #6	: CRC.TSTED,* 9295 9289 9304
20801	073450	001236			BNE 13\$	
20802	073452	104467			TRAP 67	
20803	073454	006000			ROR R0	
20804	073456	103002			BHIS 21\$	
20805	073460	000167	177172		JMP 1\$	
20806	073464	005201			INC R1	: LOOP 9230
20807	073466	020127	000004		CMP R1, #4	: LOOP,*
20808	073472	003772			BLE 20\$	
20809	073474	062706	000010		ADD #10, SP	:
20810	073500	000207			RTS PC	: 9187
20811						
20812						
20813						

: Routine Size: 215 words
: Maximum stack depth per invocation: 23 words

20815
20816
20817
20822
20823
20827
20831 073502
20832 073502 004767 177116
20833 073506 104466
20834 073510 006000
20835 073512 103773
20836 073514 000207
20837
20838
20839
20844
20845
20846 ; 9310 !<BLF/PAGE>

;ML4
;

T48::
1\$: JSR PC,\$T48 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

9307

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (99)

```

20848 :ML4
20849 :
20850 :
20851 : 9311 !
20852 : 9312 ! BGNTST
20853 : 9313 !
20854 : 9314 ! ++
20855 : 9315 ! TEST NUMBER: TST 49
20856 : 9316 !
20857 : 9317 ! TEST NAME: UNIQUE NIBBLE CRC GENERATION TEST
20858 : 9318 !
20859 : 9319 ! TEST DESCRIPTION:
20860 : 9320 ! TEST CRC CODES GENERATED FOR ONE CRC
20861 : 9321 ! GROUP (52 UNIQUE UNIBUS NIBBLES) BY:
20862 : 9322 !
20863 : 9323 ! 1. DOING MASS BUS WRITE TRANSFERSS WITH THE
20864 : 9324 ! FIRST IO BUF CRC GROUP HAVING 51 NIBBLES
20865 : 9325 ! OF XB'0000' PATTERN AND THE REMAINING NIBBLE
20866 : 9326 ! WITH PATTERNS OF XB'0000' TO XB'1111'.
20867 : 9327 !
20868 : 9328 ! 2. THEN IN DATA DIAGNOSTIC MODE CALCULATE
20869 : 9329 ! THE CRC CODE FOR THE FIRST CRC
20870 : 9330 ! GROUP AND COMPARE IT AGAINST THE
20871 : 9331 ! HARDWARE GENERATED CRC CODE.
20872 : 9332 !
20873 : 9333 ! 3. REPEAT PATTERN OF XB'0000' TO XB'1111'
20874 : 9334 ! FOR EACH OF THE 52 NIBBLES.
20875 : 9335 !
20876 : 9336 ! IMPLICIT INPUTS:
20877 : 9337 ! IO_BUF
20878 : 9338 ! A VECTOR OF 256 WORDS WHERE DATA
20879 : 9339 ! FOR MBUS READS AND WRITES TRANSFERS
20880 : 9340 ! IS FOUND.
20881 : 9341 !
20882 : 9342 !
20883 : 9343 !
20884 : 9344 !
20885 : 9345 ! local
20886 : 9346 ! DODU_FLG , !DROP UNIT FLAG
20887 : 9347 ! TEMP !TEMPORARY STORAGE LOCATION
20888 : 9348 ! NIB_SEL; !SELECTS 4 BITS 'ONE NIBBLE' FROM THE IO_BUF
20889 : 9349 ! DODU_FLG = ZEROES; !CLEAR THE DROP UNIT FLAG
20890 : 9350 !
20891 : 9351 ! incr CNT from 0 to 13 do !CLEAR THE FIRST 14 IO_BUF WORDS
20892 : 9352 ! IO_BUF [.CNT] = ZEROES;
20893 : 9353 !
20894 : 9354 ! incr WRD_CNT from 0 to 12 do !DO THIS TEST ON 13 IO_BUF WORDS
20895 : 9355 ! begin
20896 : 9356 ! NIB_SEL = -4; !RESET THE NIBBLE SELECTOR
20897 : 9357 !
20898 : 9358 ! incr NIB_TST from 0 to 3 do !DO THIS TEST ON 4 NIBBLES PER WORD
20899 : 9359 ! begin
20900 : 9360 !
20901 : 9361 ! if (.WRD_CNT eql 12) and (.NIB_TST eql 2) then exitloop; !THE LAST 2 NIBBLES ARE NOT TESTED
20902 : 9362 !

```

```

20904 :ML4
20905 :
20906 :
20907 :          9363
20908 :          9364
20909 :          9365
20910 :          9366
20911 :          9367
20912 :          9368
20913 :          9369
20914 :          9370
20915 :          9371
20916 :          9372
20917 :          9373
20918 :          9374
20919 :          9375
20920 :          9376
20921 :          9377
20922 :          9378
20923 :          9379
20924 :          9380
20925 :          9381
20926 :          9382
20927 :          9383
20928 :          9384
20929 :          9385
20930 :          9386
20931 :          9387
20932 :          9388
20933 :          9389
20934 :          9390
20935 :          9391
20936 :          9392
20937 :          9393
20938 :          9394
20939 :          9395
20940 :          9396
20941 :          9397
20942 :          9398
20943 :          9399
20944 :          9400
20945 :          9401
20946 :          9402
20947 :          9403
20948 :          9404
20949 :          9405
20950 :          9406
20951 :          9407
20952 :          9408
20956 :

```

```

NIB_SEL = .NIB_SEL + 4;
TEMP = ZEROES;

incr NIB_PAT from %b'0000' to %b'1111' by %b'0001' do
begin
  BGENSUB;
  TEMP<.NIB_SEL, 4> = .NIB_PAT;
  IO_BUF [.WRD_CNT] = .TEMP;
  CLR_MBUS;
  GD_BLK_XFER ();
  MLCS1 = write;
  TIME_OUT_LOOP;
  A_GEN = ZEROES;
  B_GEN = ZEROES;
  P_GEN = ZEROES;
  A_CAL = ZEROES;
  B_CAL = ZEROES;
  P_CAL = ZEROES;
  CAL_CRC ();

  if ERR_CHK_CRC ()
  then
  begin
    ERRDF (132, SYNC, DUMPER);
    PRINTB (THR_FMT, WRD 5, WRD 64, WRD 8);
    PRINTB (FMT_19, .B_GEN, .A_GEN, .P_GEN);
    PRINTB (FMT_20, .B_CAL, .A_CAL, .P_CAL);
    DODU_FLG = ONE;
  end;

ENDSUB;
end;

IO_BUF [.WRD_CNT] = ZEROES;
end;

end;

if .DODU_FLG IS_SET
then
begin
  DODU (.ML_LUN);
  DOCLN;
end;

ENDTST;

```

```

!INCREMENT THE NIBBLE SELECTED
!CLEAR TEMP LOCATION

!RUN PATTERNS ON THIS NIBBLE

!START OF SCOPE LOOP
!LOAD PATTERN INTO TEMP AT THIS NIBBLE
!LOAD THE GENERATED PATTERN INTO THE IO_BUF

!SET UP A GOOD BLOCK TRANSFER
!DO A WRITE TRANSFER
!WAIT FOR THE TRANSFER TO COMPLETE
!CLEAR THE GEN & CAL STORAGE LOCATIONS

!CALL ROUTINE TO CALCULATE THE CRC CODE FOR THIS PATTERN
!NOW SEE IF THE GEN CRC IS SAME AS CAL CRC
!REPORT THE ERROR IF NOT THE SAME

!END OF SCOPE LOOP

!CLEAR THIS IO_BUF WORD FO NEXT NIBBLE

!DROP THIS UNIT IF THE DODU FLAG GOT SET

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (99)

20964	073516	004167	110374		\$T49:	JSR	R1,\$\$SAVE5	:	9309
20965	073522	024646				CMP	-(SP),-(SP)	:	
20966	073524	005066	000002			CLR	2(SP)	:	DODU.FLG 9349
20967	073530	005000				CLR	R0	:	CNT 9351
20968	073532	010001			1\$:	MOV	R0,R1	:	CNT,* 9352
20969	073534	006301				ASL	R1	:	
20970	073536	005061	011600			CLR	IO.BUF(R1)	:	
20971	073542	005200				INC	R0	:	CNT 9351
20972	073544	020027	000015			CMP	R0,#15	:	CNT,*
20973	073550	003770				BLE	1\$:	
20974	073552	005002				CLR	R2	:	WRD.CNT 9354
20975	073554	012716	177774		2\$:	MOV	#-4,(SP)	:	*.NIB.SEL 9356
20976	073560	010201				MOV	R2,R1	:	WRD.CNT,* 9370
20977	073562	006301				ASL	R1	:	
20978	073564	005004				CLR	R4	:	NIB.TST 9358
20979	073566	020227	000014		3\$:	CMP	R2,#14	:	WRD.CNT,* 9361
20980	073572	001003				BNE	4\$:	
20981	073574	020427	000002			CMP	R4,#2	:	NIB.TST,*
20982	073600	001562				BEQ	8\$:	
20983	073602	062716	000004		4\$:	ADD	#4,(SP)	:	*.NIB.SEL 9363
20984	073606	005005				CLR	R5	:	TEMP 9364
20985	073610	005003				CLR	R3	:	NIB.PAT 9366
20986	073612	104402			5\$:	TRAP	2	:	9367
20987	073614	010546				MOV	R5,-(SP)	:	TEMP,* 9369
20988	073616	016646	000002			MOV	2(SP),-(SP)	:	NIB.SEL,*
20989	073622	012746	000004			MOV	#4,-(SP)	:	
20990	073626	010346				MOV	R3,-(SP)	:	NIB.PAT,*
20991	073630	004767	107446			JSR	PC,BL\$PU1	:	
20992	073634	010005				MOV	R0,R5	:	*.TEMP 9367
20993	073636	010561	011600			MOV	R5,IO.BUF(R1)	:	TEMP,* 9370
20994	073642	152777	000040	117344		BISB	#40,@ML.REG+40	:	
20995	073650	016700	117726			MOV	ML.DUT,R0	:	
20996	073654	042700	177770			BIC	#177770,R0	:	
20997	073660	142777	000007	117326		BICB	#7,@ML.REG+40	:	
20998	073666	150077	117322			BISB	R0,@ML.REG+40	:	
20999	073672	004767	122164			JSR	PC,GD.BLK.XFER	:	9372
21000	073676	012777	000061	117250		MOV	#61,@ML.REG	:	9373
21001	073704	105777	117314		6\$:	TSTB	@ML.REG+50	:	
21002	073710	100375				BPL	6\$:	
21003	073712	005067	117230			CLR	A.GEN	:	9375
21004	073716	005067	117226			CLR	B.GEN	:	9376
21005	073722	005067	117224			CLR	P.GEN	:	9377
21006	073726	005067	117206			CLR	A.CAL	:	9378
21007	073732	005067	117204			CLR	B.CAL	:	9379
21008	073736	005067	117202			CLR	P.CAL	:	9380
21009	073742	004767	120644			JSR	PC,CAL.CRC	:	9381
21010	073746	004767	121240			JSR	PC,ERR.CHK.CRC	:	9383
21011	073752	006000				ROR	R0	:	
21012	073754	103055				BCC	7\$:	
21013	073756	104455				TRAP	55	:	9386

21015										
21016				:ML4						
21017				:						
21018	073760	000204			.WORD	204				
21019	073762	010526			.WORD	SYNC				
21020	073764	024052			.WORD	DUMPER				
21021	073766	012746	006760		MOV	#WRD.8,-(SP)	:		9387	
21022	073772	012746	007224		MOV	#WRD.64,-(SP)	:			
21023	073776	012746	006330		MOV	#WRD.5,-(SP)	:			
21024	074002	012746	006034		MOV	#THR.FMT,-(SP)	:			
21025	074006	012746	000004		MOV	#4,-(SP)	:			
21026	074012	010600			MOV	SP,R0	:	SP,*		
21027	074014	104414			TRAP	14	:			
21028	074016	016716	117130		MOV	P.GEN,(SP)	:		9388	
21029	074022	016746	117120		MOV	A.GEN,-(SP)	:			
21030	074026	016746	117116		MOV	B.GEN,-(SP)	:			
21031	074032	012746	005412		MOV	#FMT.19,-(SP)	:			
21032	074036	012746	000004		MOV	#4,-(SP)	:			
21033	074042	010600			MOV	SP,R0	:	SP,*		
21034	074044	104414			TRAP	14	:			
21035	074046	016716	117072		MOV	P.CAL,(SP)	:		9389	
21036	074052	016746	117062		MOV	A.CAL,-(SP)	:			
21037	074056	016746	117060		MOV	B.CAL,-(SP)	:			
21038	074062	012746	005462		MOV	#FMT.20,-(SP)	:			
21039	074066	012746	000004		MOV	#4,-(SP)	:			
21040	074072	010600			MOV	SP,R0	:	SP,*		
21041	074074	104414			TRAP	14	:			
21042	074076	012766	000001	000044	MOV	#1,44(SP)	:	*,DODU.FLG	9390	
21043	074104	062706	000032		ADD	#32,SP	:		9385	
21044	074110	062706	000010		ADD	#10,SP	:		9367	
21045	074114	104467		7\$:	TRAP	67	:		9391	
21046	074116	006000			ROR	R0	:			
21047	074120	103634			BLO	5\$:			
21048	074122	005203			INC	R3	:	NIB.PAT	9366	
21049	074124	020327	000017		CMP	R3,#17	:	NIB.PAT,*		
21050	074130	003630			BLE	5\$:			
21051	074132	005061	011600		CLR	IO.BUF(R1)	:		9396	
21052	074136	005204			INC	R4	:	NIB.TST	9358	
21053	074140	020427	000003		CMP	R4,#3	:	NIB.TST,*		
21054	074144	003610			BLE	3\$:			
21055	074146	005202		8\$:	INC	R2	:	WRD.CNT	9354	
21056	074150	020227	000014		CMP	R2,#14	:	WRD.CNT,*		
21057	074154	003002			BGT	9\$:			
21058	074156	000167	177372		JMP	2\$:			
21059	074162	026627	000002	000001	9\$:	CMP	2(SP),#1	:	DODU.FLG,*	9401
21060	074170	001004			BNE	10\$:			
21061	074172	016700	117402		MOV	ML.LUN,R0	:		9404	
21062	074176	104451			TRAP	51	:			
21063	074200	104444			TRAP	44	:			
21064	074202	022626		10\$:	CMP	(SP)+,(SP)+	:		9309	
21065	074204	000207			RTS	PC	:			
21066										
21067										
21068										

: Routine Size: 156 words
: Maximum stack depth per invocation: 25 words

21077
21078
21082
21086 074206
21087 074206 004767 177304
21088 074212 104466
21089 074214 006000
21090 074216 103773
21091 074220 000207
21092
21093
21094
21099
21100
21101 : 9409 !
21102 : 9410 ! <BLF/PAGE>

T49::
1\$: JSR PC,ST49 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

9406

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

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (100)

```
21104 :ML4
21105 :
21106 :
21107 :          9411 !
21108 :          9412 ! BGNSTST;
21109 :          9413 !
21110 :          9414 ! ++
21111 :          9415 ! TEST NUMBER: TST 50
21112 :          9416 !
21113 :          9417 ! TEST NAME:          UNIQUE WORD CRC GENERATION TEST
21114 :          9418 !
21115 :          9419 ! TEST DESCRIPTION:
21116 :          9420 ! TEST CRC CODES GENERATED FOR ONE
21117 :          9421 ! CRC GROUP (13 UNIBUS WORDS) BY:
21118 :          9422 !
21119 :          9423 ! 1. DOING MASS BUS WRITE TRANSFERS
21120 :          9424 ! USING VARIOUS WORD DATA PATTERNES.
21121 :          9425 !
21122 :          9426 ! 2. THEN IN DATA DIAGNOSTIC MODE
21123 :          9427 ! CALCULATE THE CRC CODE FOR THE
21124 :          9428 ! FIRST CRC GROUP AND COMPARE IT TO
21125 :          9429 ! THE HARDWARE GENERATED CRC CODE.
21126 :          9430 !
21127 :          9431 ! IMPLICIT INPUTS:
21128 :          9432 ! IO_BUF
21129 :          9433 ! A VECTOR OF 256 WORDS WHERE DATA
21130 :          9434 ! FOR MBUS READS AND WRITES TRANSFERS
21131 :          9435 ! IS FOUND.
21132 :          9436 !
21133 :          9437 !
21134 :          9438 !
21135 :          9439 ! local
21136 :          9440 ! DODU_FLG,
21137 :          9441 ! TST_PAT;
21138 :          9442 !
21139 :          9443 ! DODU_FLG = ZEROES;
21140 :          9444 !
21141 :          9445 ! incr PAT_SEL from 0 to 3 do
21142 :          9446 ! begin
21143 :          9447 ! BGN SUB;
21144 :          9448 ! CLR_MBUS;
21145 :          9449 !
21146 :          9450 ! case .PAT_SEL from 0 to 3 of
21147 :          9451 ! set
21148 :          9452 !
21149 :          9453 ! [0] :
21150 :          9454 ! TST_PAT = ONES;
21151 :          9455 !
21152 :          9456 ! [1] :
21153 :          9457 ! TST_PAT = %0'052525';
21154 :          9458 !
21155 :          9459 ! [2] :
21156 :          9460 ! TST_PAT = %0'125252';
21157 :          9461 !
21158 :          9462 ! [3] :
```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (100)

```

21160 :ML4
21161 :
21162 :
21163 :          9463          TST_PAT = %'070707';
21164 :          9464          tes:
21165 :          9465
21166 :          9466          incr CNT from 0 to 255 do
21167 :          9467          IO_BUF [.CNT] = .TST_PAT;
21168 :          9468
21169 :          9469          GD_BLK_XFER ();
21170 :          9470          MLCS1 = write;
21171 :          9471          TIME_OUT_LOOP;
21172 :          9472          A_GEN = ZEROES;
21173 :          9473          B_GEN = ZEROES;
21174 :          9474          P_GEN = ZEROES;
21175 :          9475          A_CAL = ZEROES;
21176 :          9476          B_CAL = ZEROES;
21177 :          9477          P_CAL = ZEROES;
21178 :          9478          CAL_CRC ();
21179 :          9479
21180 :          9480          if ERR_CHK_CRC ()
21181 :          9481          then
21182 :          9482          begin
21183 :          9483          ERRDF (133, SYNC, DUMPER);
21184 :          9484          PRINTB (THR_FMT, WRD 5, WRD 64, WRD 8);
21185 :          9485          PRINTB (FMT_19, .B_GEN, .A_GEN, .P_GEN);
21186 :          9486          PRINTB (FMT_20, .B_CAL, .A_CAL, .P_CAL);
21187 :          9487          DODU_FLG = ONE;
21188 :          9488          end;
21189 :          9489
21190 :          9490          ENDSUB;
21191 :          9491          end;
21192 :          9492
21193 :          9493          if .DODU_FLG IS_SET
21194 :          9494          then
21195 :          9495          begin
21196 :          9496          DODU (.ML_LUN);
21197 :          9497          DOCLN;
21198 :          9498          end;
21199 :          9499
21200 :          9500          ENDTST;
21204 :
21208 074222 004167 107670          $T50:  JSR    R1,$SAVE5
21209 074226 005005                   CLR    R5
21210 074230 005004                   CLR    R4
21211 074232 104402                   $S:    TRAP  2
21212 074234 152777 000040 116752  BISB   #40,2ML.REG+40
21213 074242 016703 117334          MOV    ML.DUT,R3
    
```

!LOAD THE FIRST 13 IO_BUF WORDS WITH TST_PAT

!SET UP A GOOD BLOCK TRANSFER
!DO A WRITE TRANSFER
!WAIT FOR THE TRANSFER TO COMPLETE
!CLEAR THE GEN & CAL SAVE LOCATIONS

!CALCULATE WHAT THE CRC CODE SHOULD BE

!NOW SEE IF THE GENERATED CRC CODE IS THE SAME

!REPORT THE ERROR IF NOT THE SAME

!END OF SCOPE LOOP

!DROP THIS UNIT IF THE DODU FLAG GOT SET

```

:
: DODU.FLG
: PAT.SEL
:
    
```

```

9408
9443
9445
9446
9447
    
```

Address	OpCode	Operand 1	Operand 2	Operand 3	Comments	Line No.
21215					:ML4	
21216					:	
21217						
21218	074246	042703	177770			
21219	074252	142777	000007	116734	BIC #177770,R3	
21220	074260	150377	116730		BICB #7,@ML.REG+40	
21221	074264	010403			BISB R3,@ML.REG+40	
21222	074266	006303			MOV R4,R3 ; PAT.SEL,*	9450
21223	074270	066307	074274		ASL R3	
21224	074274	000010		3\$:	ADD 2\$(R3),PC	
21225	074276	000016			.WORD 3\$-2\$	
21226	074300	000024			.WORD 4\$-2\$	
21227	074302	000032			.WORD 5\$-2\$	
21228	074304	012701	177777	3\$:	.WORD 6\$-2\$	
21229	074310	000410			MOV #-1,R1 ; *,TST.PAT	9454
21230	074312	012701	052525	4\$:	BR 7\$;	9450
21231	074316	000405			MOV #52525,R1 ; *,TST.PAT	9457
21232	074320	012701	125252	5\$:	BR 7\$;	9450
21233	074324	000402			MOV #-52526,R1 ; *,TST.PAT	9460
21234	074326	012701	070707	6\$:	BR 7\$;	9450
21235	074332	005002		7\$:	MOV #70707,R1 ; *,TST.PAT	9463
21236	074334	010203		8\$:	CLR R2 ; CNT	9466
21237	074336	006303			MOV R2,R3 ; CNT,*	9467
21238	074340	010163	011600		ASL R3	
21239	074344	005202			MOV R1,IO.BUF(R3) ; TST.PAT,*	
21240	074346	020227	000377		INC R2 ; CNT	9466
21241	074352	003770			CMP R2,#377 ; CNT,*	
21242	074354	004767	121502		BLE 8\$	
21243	074360	012777	000061	116566	JSR PC,GD.BLK.XFER ;	9469
21244	074366	105777	116632	9\$:	MOV #61,@ML.REG ;	9470
21245	074372	100375			TSTB @ML.REG+50	
21246	074374	005067	116546		BPL 9\$	
21247	074400	005067	116544		CLR A.GEN ;	9472
21248	074404	005067	116542		CLR B.GEN ;	9473
21249	074410	005067	116524		CLR P.GEN ;	9474
21250	074414	005067	116522		CLR A.CAL ;	9475
21251	074420	005067	116520		CLR B.CAL ;	9476
21252	074424	004767	120162		CLR P.CAL ;	9477
21253	074430	004767	120556		JSR PC,CAL.CRC ;	9478
21254	074434	006000			JSR PC,ERR.CHK.CRC ;	9480
21255	074436	103054			ROR R0	
21256	074440	104455			BCC 10\$	
21257	074442	000205			TRAP 55 ;	9483
21258	074444	010526			.WORD 205	
21259	074446	024052			.WORD SYNC	
21260	074450	012746	006360		.WORD DUMPER	
21261	074454	012746	007224		MOV #WRD.8,-(SP) ;	9484
21262	074460	012746	006330		MOV #WRD.64,-(SP)	
21263	074464	012746	006034		MOV #WRD.5,-(SP)	
21264	074470	012746	000004		MOV #THR.FMT,-(SP)	
21265	074474	010600			MOV #4,-(SP)	
21266	074476	104414			MOV SP,R0 ; SP,*	
21267	074500	016716	116446		TRAP 14 ;	9485
21268	074504	016746	116436		MOV P.GEN,(SP)	
21269	074510	016746	116434		MOV A.GEN,-(SP)	
					MOV B.GEN,-(SP)	


```

21271      :ML4
21272      :
21273
21274 074514 012746 005412      MOV      #FMT.19, -(SP)
21275 074520 012746 000004      MOV      #4, -(SP)
21276 074524 010600              MOV      SP, R0      ; SP,*
21277 074526 104414              TRAP     14
21278 074530 016716 116410      MOV      P.CAL, (SP)      ;
21279 074534 016746 116400      MOV      A.CAL, -(SP)      ;
21280 074540 016746 116376      MOV      B.CAL, -(SP)
21281 074544 012746 005462      MOV      #FMT.20, -(SP)
21282 074550 012746 000004      MOV      #4, -(SP)
21283 074554 010600              MOV      SP, R0      ; SP,*
21284 074556 104414              TRAP     14
21285 074560 012705 000001      MOV      #1, R5      ; *,DODU.FLG
21286 074564 062706 000032      ADD      #32, SP      ;
21287 074570 104467 10$:      TRAP     67      ;
21288 074572 006000              ROR      R0
21289 074574 103616              BLO      1$
21290 074576 005204              INC      R4      ; PAT.SEL
21291 074600 020427 000003      CMP      R4, #3      ; PAT.SEL,*
21292 074604 003612              BLE      1$
21293 074606 005305              DEC      R5      ; DODU.FLG
21294 074610 001004              BNE      11$
21295 074612 016700 116762      MOV      ML.LUN, R0      ;
21296 074616 104451              TRAP     51
21297 074620 104444              TRAP     44
21298 074622 000207 11$:      RTS      PC      ;
21299
21300      ; Routine Size: 129 words
21301      ; Maximum stack depth per invocation: 19 words
21306
21307
21311
21315 074624      T50::
21316 074624 004767 177372 1$:      JSR      PC, $T50      ;
21317 074630 104466              TRAP     66
21318 074632 006000              ROR      R0
21319 074634 103773              BLO      1$
21320 074636 000207              RTS      PC
21321
21322      ; Routine Size: 6 words
21323      ; Maximum stack depth per invocation: 0 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

9486

9487

9482

9488

9445

9493

9496

9408

9498

21332
21333
21334 :

9501 !<BLF/PAGE>

21336 :ML4
21337 :
21338 :
21339 :
21340 :
21341 :
21342 :
21343 :
21344 :
21345 :
21346 :
21347 :
21348 :
21349 :
21350 :
21351 :
21352 :
21353 :
21354 :
21355 :
21356 :
21357 :
21358 :
21359 :
21360 :
21361 :
21362 :
21363 :
21364 :
21365 :
21366 :
21367 :
21368 :
21369 :
21370 :
21371 :
21372 :
21373 :
21374 :
21375 :
21376 :
21377 :
21378 :
21379 :
21380 :
21381 :
21382 :
21383 :
21384 :
21385 :
21386 :
21387 :
21388 :
21389 :
21390 :

9502 !
9503 BGNTST;
9504
9505 !++
9506
9507
9508
9509
9510
9511
9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522
9523
9524
9525
9526
9527
9528
9529
9530
9531
9532
9533
9534
9535
9536
9537
9538
9539
9540
9541
9542
9543
9544
9545
9546
9547
9548
9549
9550
9551
9552
9553

BGNTST;

!++

TEST NUMBER: TST 51

TEST NAME: CORRECTABLE ERROR SYNDROME DECODE TEST

TEST DESCRIPTION:

TEST SYNDROME DECODE AND ERROR
CORRECTION TO DECODE AND CORRECT
SINGLE BIT AND MULTIPLE BIT
CHANNEL ERRORS BY:

1. VIA ECC DIAG MODE FORCE SYNDROME
BITS TO DECODE SINGLE BIT AND
MULTIPLE BIT CHANNEL ERRORS.
2. THEN DO A MASS BUS READ TRANSFER
AND EXAM THE IO BUF FOR CORRECT
BIT COMPLIMENTING.

IMPLICIT INPUTS:

IO_BUF
A VECTOR OF 256 WORDS WHERE DATA
FOR MBUS READS AND WRITES TRANSFERS
IS FOUND.

local

BITS_XFERED;

BAI = ONE;

IO_BUF = ZEROES;

GD_BLK_XFER ();

ML[SI] = write;

TIME_OUT LOOP;

BITS_XFERED = -1;

incr PLOG from 0 to 5 do
begin

incr CHANNEL from 0 to 35 do

begin

BGNSUB;

CLR MBUS;

BITS_XFERED = .BITS_XFERED + 1;

incr CNT from 0 to 255 do
IO_BUF [.CNT] = ZEROES;

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (101)

!POINTS TO THE CHANNEL BEING TESTED

!BEFORE WE START LETS CLEAR OUT THE ML11'S
!GOOD BLOCK BY WRITTING ZEROES TO IT

!RESET THE CHANNEL POINTER

!TEST ONE CRC GROUP '6 ARRAY WORDS'

!TEST 36 BITS IN EACH ARRAY WORD

!START OF SCOPE LOOP

!CLEAR THE DRIVE

!INCREMENT THE CHANNEL POINTER

!CLEAR THE FIRST CRC GROUP IN THE IO_BUF

```

21392 :ML4
21393 :
21394 :
21395 : 9554
21396 : 9555 FORCE REM (.PLOG, .CHANNEL); !FORCE THIS BIT IN THIS WORD IN ERROR
21397 : 9556 GD BLR XFER (); !SET UP A GOOD BLOCK TRANSFER
21398 : 9557 ECC DIS = ZERO; !CLEAR ECC DISABLE
21399 : 9558 MLC51 = read; !DO A READ TRANSFER
21400 : 9559 TIME_OUT_LOOP; !WAIT FOR THE TRANSFER TO COMPLETE
21401 : 9560
21402 : 9561 if not FIND_COMP_BIT (.BITS_XFERED) !SEARCH THE IO_BUF FOR THE COMP BIT
21403 : 9562 then
21404 : 9563 begin !REPORT AN ERROR IF THIS BIT IS NOT COMP
21405 : 9564 ERRDF (134, SYNC, DUMPER);
21406 : 9565 PRINTB (THR_FMT, WRD 74, WRD 75, PHR_1);
21407 : 9566 PRINTB (FMT_21, .PLOG, .CHANNEL);
21408 : 9567 end;
21409 : 9568
21410 : 9569 incr WRD_CNT from 0 to 12 do !SEE IF ANY OTHER BITS GOT COMP'ED
21411 : 9570
21412 : 9571 if .IO_BUF [.WRD_CNT] neq ZEROES !TEST THE IO_BUF FOR ZEROES
21413 : 9572 then
21414 : 9573 begin !REPORT AN ERROR IF ANY OTHERS ARE SET
21415 : 9574 ERRDF (135, SYNC, DUMPER);
21416 : 9575 PRINTB (FOR_FMT, WRD 61, WRD 74, WRD 75, PHR_5);
21417 : 9576 PRINTB (FMT_21, .PLOG, .CHANNEL);
21418 : 9577 end;
21419 : 9578
21420 : 9579 ENDSUB; !END OF SCOPE LOOP
21421 : 9580 end;
21422 : 9581
21423 : 9582 end;
21424 : 9583
21425 : 9584 BITS_XFERED = -1; !RESET THE CHANNEL POINTER
21426 : 9585
21427 : 9586 incr CHANNEL from 0 to 35 do !TEST 36 CHANNELS FOR MULTIPLE BIT ERRORS
21428 : 9587 begin
21429 : 9588 BGN SUB; !START OF SCOPE LOOP
21430 : 9589 CLR MBUS; !CLEAR THE DRIVE
21431 : 9590 BITS_XFERED = .BITS_XFERED + 1; !INCREMENT THE CHANNEL POINTER
21432 : 9591
21433 : 9592 incr CNT from 0 to 20 do !CLEAR THE FIRST CRC GROUP IN THE IO_BUF
21434 : 9593 IO_BUF [.CNT] = ZEROES;
21435 : 9594
21436 : 9595 FORCE REM (58, .CHANNEL); !FORCE ALL 6 BITS IN THIS CHANNEL IN ERROR
21437 : 9596 GD BLR XFER (); !SET UP A GOOD BLOCK TRANSFER
21438 : 9597 ECC DIS = ZERO; !CLEAR ECC DISABLE
21439 : 9598 MLC51 = read; !DO A READ TRANSFER
21440 : 9599 TIME_OUT_LOOP; !WAIT FOR THE TRANSFER TO COMPLETE
21441 : 9600
21442 : 9601 incr BIT_OFFSET from 0 to 180 by 36 do !SEE IF EVERY 36TH BIT GOT COMP'ED
21443 : 9602
21444 : 9603 if not FIND_COMP_BIT (.BITS_XFERED + .BIT_OFFSET) !SEARCH THE IO_BUF FOR THE COMP'ED BITS
21445 : 9604 then
21446 : 9605 begin !REPORT AN ERROR IF ANY OF THE BITS ARE NOT COMP'ED

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (101)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (101)

```

21448 :ML4
21449 :
21450 :
21451 :          9606          ERRDF (134, SYNC, DUMPER);
21452 :          9607          PRINTB (THR_FMT, WRD 74, WRD 75, PHR_1);
21453 :          9608          PRINTB (FMT_21, 58, CHANNEL);
21454 :          9609          end;
21455 :          9610
21456 :          9611          incr WRD_CNT from 0 to 12 do          !ALSO SEE IF ANY OTHER BIT GOT COMP'ED
21457 :          9612
21458 :          9613          if .IO_BUF [.WRD_CNT] neq ZEROES          !TEST THE IO_BUF FOR ALL ZEROES
21459 :          9614          then
21460 :          9615          begin          !REPORT AN ERROR IF ANY OTHERS ARE COMP'ED
21461 :          9616          ERRDF (135, SYNC, DUMPER);
21462 :          9617          PRINTB (FIV_FMT, WRD 61, WRD 74, WRD_75, WRD_12, WRD_67);
21463 :          9618          PRINTB (FMT_21, 58, CHANNEL);
21464 :          9619          end;
21465 :          9620
21466 :          9621          ENDSUB;          !END OF SCOPE LOOP
21467 :          9622          end;
21468 :          9623
21469 :          9624          ENDTST;
21473 :
21477 074640 004167 107252          $T51:  JSR      R1, $SAVES          :
21478 074644 152777 000010 116342  BISH  #10, @ML.REG+40          :
21479 074652 005067 114722          CLR      IO_BUF          :
21480 074656 004767 121200          JSR      PC, GD.BLK.XFER          :
21481 074662 012777 000061 116264  MOV      #61, @ML.REG          :
21482 074670 105777 116330          1$:    TSTB  @ML.REG+50          :
21483 074674 100375          BPL      1$          :
21484 074676 012705 177777          MOV      #-1, R5          : * ,BITS.XFERED
21485 074702 005004          CLR      R4          : PLOG
21486 074704 005003          2$:    CLR      R3          : CHANNEL
21487 074706 104402          3$:    TRAP  2          :
21488 074710 152777 000040 116276  BISH  #40, @ML.REG+40          :
21489 074716 016702 116660          MOV      ML.DUT, R2          :
21490 074722 042702 177770          BIC      #177770, R2          :
21491 074726 142777 000007 116260  BICB  #7, @ML.REG+40          :
21492 074734 150277 116254          BISH  R2, @ML.REG+40          :
21493 074740 005205          INC      R5          : BITS.XFERED
21494 074742 005001          CLR      R1          : CNT
21495 074744 010102          4$:    MOV      R1, R2          : CNT,*
21496 074746 006302          ASL      R2          :
21497 074750 005062 011600          CLR      IO_BUF (R2)          :
21498 074754 005201          INC      R1          : CNT
21499 074756 020127 000377          CMP      R1, #377          : CNT,*
21500 074762 003770          BLE      4$          :
21501 074764 010446          MOV      R4, -(SP)          : PLOG,*

```

Address	Hex	Hex	Hex	Label	Code	Comment	Date/Time	Page
21503							22-Dec-1980 09:24:31	TOPS
21504							22-Dec-1980 09:21:22	PA:<
21505								
21506	074766	010346			MOV	R3,-(SP) ; CHANNEL,*		
21507	074770	004767	120606		JSR	PC,FORCE.REM		
21508	074774	004767	121062		JSR	PC,GD.BLK.XFER		9556
21509	075000	142777	000002	116266	BICB	#2,AML.REG+120		9557
21510	075006	012777	000071	116140	MOV	#71,AML.REG		9558
21511	075014	105777	116204	5S:	TSTB	AML.REG+50		
21512	075020	100375			BPL	5S		
21513	075022	010546			MOV	R5,-(SP) ; BITS.XFERED,*		9561
21514	075024	004767	120400		JSR	PC,FIND.COMP.BIT		
21515	075030	005726			TST	(SP)+		
21516	075032	006000			ROR	R0		
21517	075034	103432			BLO	6S		
21518	075036	104455			TRAP	55 ;		9564
21519	075040	000206			.WORD	206		
21520	075042	010526			.WORD	SYNC		
21521	075044	024052			.WORD	DUMPER		
21522	075046	012746	007366		MOV	#PHR.1 -(SP) ;		9565
21523	075052	012746	007330		MOV	#WRD.75 -(SP)		
21524	075056	012746	007316		MOV	#WRD.74 -(SP)		
21525	075062	012746	006034		MOV	#THR.FMT -(SP)		
21526	075066	012746	000004		MOV	#4 -(SP)		
21527	075072	010600			MOV	SP,R0 ; SP,*		
21528	075074	104414			TRAP	14		
21529	075076	010316			MOV	R3,(SP) ; CHANNEL,*		9566
21530	075100	010446			MOV	R4,-(SP) ; PLOG,*		
21531	075102	012746	005534		MOV	#FMT.21 -(SP)		
21532	075106	012746	000003		MOV	#3 -(SP)		
21533	075112	010600			MOV	SP,R0 ; SP,*		
21534	075114	104414			TRAP	14		
21535	075116	062706	000020		ADD	#20,SP		9563
21536	075122	005001		6S:	CLR	R1 ; WRD.CNT		9569
21537	075124	010102		7S:	MOV	R1,R2 ; WRD.CNT,*		9571
21538	075126	006302			ASL	R2		
21539	075130	005762	011600		TST	IO.BUF(R2)		
21540	075134	001434			BEQ	8S		
21541	075136	104455			TRAP	55 ;		9574
21542	075140	000207			.WORD	207		
21543	075142	010526			.WORD	SYNC		
21544	075144	024052			.WORD	DUMPER		
21545	075146	012746	007472		MOV	#PHR.5 -(SP) ;		9575
21546	075152	012746	007330		MOV	#WRD.75 -(SP)		
21547	075156	012746	007316		MOV	#WRD.74 -(SP)		
21548	075162	012746	007174		MOV	#WRD.61 -(SP)		
21549	075166	012746	006046		MOV	#FOR.FMT -(SP)		
21550	075172	012746	000005		MOV	#5 -(SP)		
21551	075176	010600			MOV	SP,R0 ; SP,*		
21552	075200	104414			TRAP	14		
21553	075202	010316			MOV	R3,(SP) ; CHANNEL,*		9576
21554	075204	010446			MOV	R4,-(SP) ; PLOG,*		
21555	075206	012746	005534		MOV	#FMT.21 -(SP)		
21556	075212	012746	000003		MOV	#3 -(SP)		
21557	075216	010600			MOV	SP,R0 ; SP,*		


```

21615      :ML4
21616      :
21617      :
21618 075452 012746 007316      MOV      #WRD.74,-(SP)
21619 075456 012746 006034      MOV      #THR.FMT,-(SP)
21620 075462 012746 000004      MOV      #4,-(SP)
21621 075466 010600      MOV      SP,R0      ; SP,*
21622 075470 104414      TRAP     14
21623 075472 010316      MOV      R3,(SP)      ; CHANNEL,*      9608
21624 075474 012746 000072      MOV      #72,-(SP)
21625 075500 012746 005534      MOV      #FMT.21,-(SP)
21626 075504 012746 000003      MOV      #3,-(SP)
21627 075510 010600      MOV      SP,R0      ; SP,*
21628 075512 104414      TRAP     14
21629 075514 062706 000020      ADD      #20,SP      ;      9605
21630 075520 062702 000044      ADD      #44,R2      ; * ,BIT.OFFSET      9601
21631 075524 020227 000264      CMP      R2,#264      ; BIT.OFFSET,*
21632 075530 003731      BLE      12$
21633 075532 005001      CLR      R1      ; WRD.CNT      9611
21634 075534 010102      MOV      R1,R2      ; WRD.CNT,*      9613
21635 075536 006302      ASL      R2
21636 075540 005762 011600      TST      IO.BUF(R2)
21637 075544 001437      BEQ      15$
21638 075546 104455      TRAP     55      ;      9616
21639 075550 000207      .WORD   207
21640 075552 010526      .WORD   SYNC
21641 075554 024052      .WORD   DUMPER
21642 075556 012746 007240      MOV      #WRD.67,-(SP)      ;      9617
21643 075562 012746 006426      MOV      #WRD.12,-(SP)
21644 075566 012746 007330      MOV      #WRD.75,-(SP)
21645 075572 012746 007316      MOV      #WRD.74,-(SP)
21646 075576 012746 007174      MOV      #WRD.61,-(SP)
21647 075602 012746 006062      MOV      #FIV.FMT,-(SP)
21648 075606 012746 000006      MOV      #6,-(SP)
21649 075612 010600      MOV      SP,R0      ; SP,*
21650 075614 104414      TRAP     14
21651 075616 010316      MOV      R3,(SP)      ; CHANNEL,*      9618
21652 075620 012746 000072      MOV      #72,-(SP)
21653 075624 012746 005534      MOV      #FMT.21,-(SP)
21654 075630 012746 000003      MOV      #3,-(SP)
21655 075634 010600      MOV      SP,R0      ; SP,*
21656 075636 104414      TRAP     14
21657 075640 062706 000024      ADD      #24,SP      ;      9615
21658 075644 005201      INC      R1      ; WRD.CNT      9611
21659 075646 020127 000014      CMP      R1,#14      ; WRD.CNT,*
21660 075652 003730      BLE      14$
21661 075654 022626      CMP      (SP)+,(SP)+      ;      9587
21662 075656 104467      TRAP     67      ;      9619
21663 075660 006000      ROR      R0
21664 075662 103604      BLO      9$
21665 075664 005203      INC      R3      ; CHANNEL      9586
21666 075666 020327 000043      CMP      R3,#43      ; CHANNEL,*
21667 075672 003600      BLE      9$
21668 075674 000207      RTS
21669      PC      ;      9500

```


22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
21671      ;ML4
21672      ;
21673
21674      ; Routine Size: 271 words
21675      ; Maximum stack depth per invocation: 18 words
21680
21681
21685
21689 075676
21690 075676 004767 176736
21691 075702 104466
21692 075704 006000
21693 075706 103773
21694 075710 000207
21695
21696      ; Routine Size: 6 words
21697      ; Maximum stack depth per invocation: 0 words
21702
21703
21704 ;      9625 !<BLF/PAGE>
```

```
T51::
1$:      JSR      PC,$T51
          TRAP    66
          ROR     R0
          BLO    1$
          RTS     PC
```

9622

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (102)

21706 :ML4
21707 :
21708 :
21709 :
21710 :
21711 :
21712 :
21713 :
21714 :
21715 :
21716 :
21717 :
21718 :
21719 :
21720 :
21721 :
21722 :
21723 :
21724 :
21725 :
21726 :
21727 :
21728 :
21729 :
21730 :
21731 :
21732 :
21733 :
21734 :
21735 :
21736 :
21737 :
21738 :
21739 :
21740 :
21741 :
21742 :
21743 :
21744 :
21745 :
21746 :
21747 :
21748 :
21749 :
21750 :
21751 :
21752 :
21753 :
21754 :
21755 :
21756 :
21757 :
21758 :
21759 :
21760 :

9626
9627
9628
9629
9630
9631
9632
9633
9634
9635
9636
9637
9638
9639
9640
9641
9642
9643
9644
9645
9646
9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670
9671
9672
9673
9674
9675
9676
9677

```

!
BGNTST;
!++
TEST NUMBER: TST 52
TEST NAME:          UNCORRECTABLE ERROR SYNDROME DECODE TEST AT CHANNELS > 35
TEST DESCRIPTION:
TEST SYNDROME DECODE TO DETECT BUT
NOT CORRECT UNCORRECTABLE CHANNEL
ERRORS BY:
1. VIA ECC DIAG MODE FORCE SYNDROME
   BITS TO INDICATE CHANNEL ERRORS
   AT WORDS OF A CRC GROUP BUT AT CHANNELS
   GREATER THAN 35.
2. DO A MASS BUS READ TRANSFER
3. THEN EXAMIN THE IO BUF FOR NO
   BITS COMPLIMENTED AND ECH AND
   BITS SET
IMPLICIT INPUTS:
IO_BUF
A VECTOR OF 256 WORDS WHERE DATA
FOR MBUS READS AND WRITES TRANSFERS
IS FOUND.
--
Local
PLOG;
BAI = ONE;
IO_BUF = ZEROES;
GD_BLK_XFER ();
MLCS1 = write;
TIME_OUT_LOOP;
incr PLOG_SEL from 0 to 6 do
begin
case .PLOG_SEL from 0 to 6 of
set
[0] :
PLOG = 0;
[1] :

```

```

!INDEX INTO REMAINDER TABLE 'REM_TBL'
!FIRST LETS CLEAR THE ML11'S GOOD BLOCK
!BY WRITING ZEROES TO IT
!TEST SYNDROME DECODE AT 7 DIFFERENT PLOG VALUES
!SELECT A PLOG VALUE
!POINTS TO WORD 0 IN THE CRC GROUP
!POINTS TO WORD 1 IN THE CRC GROUP

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (102)

```

21762 :ML4
21763 :
21764 :
21765 :          9678          PLOG = 1;
21766 :          9679
21767 :          9680          [2] :          !POINTS TO WORD 2 IN THE CRC GROUP
21768 :          9681          PLOG = 2;
21769 :          9682
21770 :          9683          [3] :          !POINTS TO WORD 3 IN THE CRC GROUP
21771 :          9684          PLOG = 3;
21772 :          9685
21773 :          9686          [4] :          !POINTS TO WORD 4 IN THE CRC GROUP
21774 :          9687          PLOG = 4;
21775 :          9688
21776 :          9689          [5] :          !POINTS TO WORD 5 IN THE CRC GROUP
21777 :          9690          PLOG = 5;
21778 :          9691
21779 :          9692          [6] :          !POINTS TO ALL 6 WORDS IN THE CRC GROUP
21780 :          9693          tes:
21781 :          9694
21782 :          9695          incr CHANNEL from 36 to 62 do          !TEST AT CHANNELS > 35 TO FORCE UNCORRECTABLE ERROR
21783 :          9696          begin
21784 :          9697          BGN SUB;          !START OF SCOPE LOOP
21785 :          9698          CLR_MBUS;          !CLEAR THE DRIVE
21786 :          9699
21787 :          9700          incr COUNT from 0 to 255 do          !CLEAR THE FIRST CRC GROUP IN THE IO_BUF
21788 :          9701          IO_BUF [.COUNT] = ZEROES;
21789 :          9702
21790 :          9703          FORCE REM (.PLOG, .CHANNEL);          !FORCE UNCORRECTABLE ERROR AT THIS WORD AND CHANNEL
21791 :          9704          GD BLR_XFER ();          !SET UP A GOOD BLOCK TRANSFER
21792 :          9705          ECC_DIS = ZERO;          !CLEAR ECC DISABLE
21793 :          9706          MLCS1 = read;          !DO A READ TRANSFER
21794 :          9707          TIME_OUT_LOOP;          !WAIT FOR THE TRANSFER TO COMPLETE
21795 :          9708
21796 :          9709          if not ((.ECH_ERR) and (.UNC_ERR))          !SEE IF THE ERROR WAS DETECTED
21797 :          9710          then
21798 :          9711          begin          !ERROR IF NOT DETECTED
21799 :          9712          ERRDF (136, SYNC, DUMPER);
21800 :          9713          PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_76, WRD_9);
21801 :          9714          PRINTB (FMT_21, .PLOG, .CHANNEL);
21802 :          9715          end;
21803 :          9716
21804 :          9717          incr CNT from 0 to 12 do          !SEE IF ANY CORRECTION WAS DONE
21805 :          9718
21806 :          9719          if .IO_BUF [.CNT] neq ZEROES          !SEARCH THE IO_BUF FOR ALL ZEROES
21807 :          9720          then
21808 :          9721          begin          !REPORT AN ERROR IF ANY CORRECTION WAS DONE
21809 :          9722          ERRDF (137, SYNC, DUMPER);
21810 :          9723          PRINTB (FIV_FMT, WRD_61, WRD_74, WRD_75, WRD_12, WRD_67);
21811 :          9724          PRINTB (FMT_21, .PLOG, .CHANNEL);
21812 :          9725          end;
21813 :          9726
21814 :          9727          ENDSUB;
21815 :          9728          !END OF SCOPE LOOP
21816 :          9729          end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (102)

```

21818 ;ML4
21819 ;
21820 ;
21821 : 9730
21822 : 9731      end:
21823 : 9732
21824 : 9733      ENDTST:
21828 ;
21832 075712 004167 106200 $T52: JSR R1,$SAVE5 : 9624
21833 075716 152777 000010 115270 BISS #10,@ML.REG+40 : 9662
21834 075724 005067 113650 CLR IO.BUF : 9663
21835 075730 004767 120126 JSR PC,GD.BLK.XFER : 9664
21836 075734 012777 000061 115212 MOV #61,@ML.REG : 9665
21837 075742 105777 115256 1$: TSTB @ML.REG+50
21838 075746 100375 BPL 1$
21839 075750 005005 CLR R5 : PLOG.SEL 9668
21840 075752 010504 2$: MOV R5,R4 : PLOG.SEL,* 9671
21841 075754 006304 ASL R4
21842 075756 066407 075762 ADD 3$(R4),PC
21843 075762 000016 3$: .WORD 4$-3$
21844 075764 000022 .WORD 5$-3$
21845 075766 000030 .WORD 6$-3$
21846 075770 000036 .WORD 7$-3$
21847 075772 000044 .WORD 8$-3$
21848 075774 000052 .WORD 9$-3$
21849 075776 000060 .WORD 10$-3$
21850 076000 005003 4$: CLR R3 : PLOG 9675
21851 076002 000421 BR 11$ : 9671
21852 076004 012703 000001 5$: MOV #1,R3 : *,PLOG 9678
21853 076010 000416 BR 11$ : 9671
21854 076012 012703 000002 6$: MOV #2,R3 : *,PLOG 9681
21855 076016 000413 BR 11$ : 9671
21856 076020 012703 000003 7$: MOV #3,R3 : *,PLOG 9684
21857 076024 000410 BR 11$ : 9671
21858 076026 012703 000004 8$: MOV #4,R3 : *,PLOG 9687
21859 076032 000405 BR 11$ : 9671
21860 076034 012703 000005 9$: MOV #5,R3 : *,PLOG 9690
21861 076040 000402 BR 11$ : 9671
21862 076042 012703 000072 10$: MOV #72,R3 : *,PLOG 9693
21863 076046 012704 000044 11$: MOV #44,R4 : *,CHANNEL 9696
21864 076052 104402 12$: TRAP 2 : 9697
21865 076054 152777 000040 115132 BISS #40,@ML.REG+40 : 9698
21866 076062 016702 MOV ML,DUT,R2
21867 076066 042702 BIC #177770,R2
21868 076072 142777 000007 115114 BICB #7,@ML.REG+40
21869 076100 150277 115110 BISS R2,@ML.REG+40
21870 076104 005001 CLR R1 : COUNT 9701
21871 076106 010102 13$: MOV R1,R2 : COUNT,* 9702

```

21873						:ML4				
21874						:				
21875										
21876	076110	006302					ASL	R2		
21877	076112	005062	011600				CLR	I0.BUF(R2)		
21878	076116	005201					INC	R1	: COUNT	9701
21879	076120	020127	000377				CMP	R1,#377	: COUNT,*	
21880	076124	003770					BLE	13\$		
21881	076126	010346					MOV	R3,-(SP)	: PLOG,*	9704
21882	076130	010446					MOV	R4,-(SP)	: CHANNEL,*	
21883	076132	004767	117444				JSR	PC,FORCE.REM		
21884	076136	004767	117720				JSR	PC,GD.BLK.XFER		9705
21885	076142	142777	000002	115124			BICB	#2,@ML.REG+120		9706
21886	076150	012777	000071	114776			MOV	#71,@ML.REG		9707
21887	076156	105777	115042			14\$:	TSTB	@ML.REG+50		
21888	076162	100375					BPL	14\$		
21889	076164	132777	000100	115042			BITB	#100,@ML.REG+60	:	9710
21890	076172	001403					BEQ	15\$		
21891	076174	005777	115164				TST	@ML.REG+210		
21892	076200	100434					BMI	16\$		
21893	076202	104455				15\$:	TRAP	55	:	9713
21894	076204	000210					.WORD	210		
21895	076206	010526					.WORD	SYNC		
21896	076210	024052					.WORD	DUMPER		
21897	076212	012746	006374				MOV	#WORD.9,-(SP)	:	9714
21898	076216	012746	007340				MOV	#WORD.76,-(SP)		
21899	076222	012746	006406				MOV	#WORD.10,-(SP)		
21900	076226	012746	007240				MOV	#WORD.67,-(SP)		
21901	076232	012746	006046				MOV	#FOR.FMT,-(SP)		
21902	076236	012746	000005				MOV	#5,-(SP)		
21903	076242	010600					MOV	SP,R0	: SP,*	
21904	076244	104414					TRAP	14		
21905	076246	010416					MOV	R4,(SP)	: CHANNEL,*	9715
21906	076250	010346					MOV	R3,-(SP)	: PLOG,*	
21907	076252	012746	005534				MOV	#FMT.21,-(SP)		
21908	076256	012746	000003				MOV	#3,-(SP)		
21909	076262	010600					MOV	SP,R0	: SP,*	
21910	076264	104414					TRAP	14		
21911	076266	062706	000022				ADD	#22,SP	:	9712
21912	076272	005001				16\$:	CLR	R1	: CNT	9718
21913	076274	010102				17\$:	MOV	R1,R2	: CNT,*	9720
21914	076276	006302					ASL	R2		
21915	076300	005762	011600				TST	I0.BUF(R2)		
21916	076304	001436					BEQ	18\$		
21917	076306	104455					TRAP	55	:	9723
21918	076310	000211					.WORD	211		
21919	076312	010526					.WORD	SYNC		
21920	076314	024052					.WORD	DUMPER		
21921	076316	012746	007240				MOV	#WORD.67,-(SP)	:	9724
21922	076322	012746	006426				MOV	#WORD.12,-(SP)		
21923	076326	012746	007330				MOV	#WORD.75,-(SP)		
21924	076332	012746	007316				MOV	#WORD.74,-(SP)		
21925	076336	012746	007174				MOV	#WORD.61,-(SP)		
21926	076342	012746	006062				MOV	#FIV.FMT,-(SP)		
21927	076346	012746	000006				MOV	#6,-(SP)		

```

21929      ;ML4
21930      ;
21931
21932 076352 010600      MOV      SP,R0      ; SP,*
21933 076354 104414      TRAP     14
21934 076356 010416      MOV      R4,(SP)    ; CHANNEL,*
21935 076360 010346      MOV      R3,-(SP)   ; PLOG,*
21936 076362 012746 005534  MOV      #FMT.21,-(SP)
21937 076366 012746 000003  MOV      #3,-(SP)
21938 076372 010600      MOV      SP,R0      ; SP,*
21939 076374 104414      TRAP     14
21940 076376 062706 000024  ADD      #24,SP
21941 076402 005201 18$: INC      R1      ; CNT
21942 076404 020127 000014  CMP      R1,#14    ; CNT,*
21943 076410 003731      BLE     17$
21944 076412 022626      CMP      (SP)+,(SP)+
21945 076414 104467      TRAP     67
21946 076416 006000      ROR     R0
21947 076420 103614      BLO     12$
21948 076422 005204      INC     R4      ; CHANNEL
21949 076424 020427 000076  CMP      R4,#76    ; CHANNEL,*
21950 076430 003610      BLE     12$
21951 076432 005205      INC     R5      ; PLOG.SEL
21952 076434 020527 000006  CMP      R5,#6     ; PLOG.SEL,*
21953 076440 003002      BGT     19$
21954 076442 000167 177304  JMP     2$
21955 076446 000207 19$: RTS     PC
21956
21957      ; Routine Size: 175 words
21958      ; Maximum stack depth per invocation: 18 words
21963
21964
21968
21972 076450      T52::
21973 076450 004767 177236 1$: JSR     PC,$T52
21974 076454 104466      TRAP     66
21975 076456 006000      ROR     R0
21976 076460 103773      BLO     1$
21977 076462 000207      RTS     PC
21978
21979      ; Routine Size: 6 words
21980      ; Maximum stack depth per invocation: 0 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

9725

9722
9718

9697
9726

9696

9668

9624

9731

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 473^{F 4}

SEQ 0460

21989
21990
21991 ; 9734 !<BLF/PAGE>

21993 :ML4

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (103)

```

21996 : 9735 !
21997 : 9736 BGNTST;
21998 : 9737
21999 : 9738 !++
22000 : 9739 TEST NUMBER: TST 53
22001 : 9740
22002 : 9741 TEST NAME: MULTIPLE CHANNEL ERROR SYNDROME DECODE TEST
22003 : 9742
22004 : 9743 TEST DESCRIPTION:
22005 : 9744 TEST SYNDROME DECODE TO DETECT BUT NOT
22006 : 9745 CORRECT UNCORRECTABLE MULTIPLE CHANNEL
22007 : 9746 ERRORS BY:
22008 : 9747
22009 : 9748 1. VIA ECC DIAGNOSTIC MODE FORCE SYNDROME
22010 : 9749 BITS TO INDICATE MULTIPLE CHANNEL ERRORS
22011 : 9750 BY ASSIGNING CRC_A EQUAL TO CRC_B
22012 : 9751
22013 : 9752 2. DO A MASS BUS READ TRANSFER
22014 : 9753
22015 : 9754 3. THEN EXAMIN THE IO BUF FOR NO BITS
22016 : 9755 COMPLIMENTED AND ECH AND UNC BITS SET
22017 : 9756
22018 : 9757 IMPLICIT INPUTS:
22019 : 9758 IO_BUF
22020 : 9759 A VECTOR OF 256 WORDS WHERE DATA
22021 : 9760 FOR MBUS READS AND WRITES TRANSFERS
22022 : 9761 IS FOUND.
22023 : 9762
22024 : 9763
22025 : 9764 !--
22026 : 9765
22027 : 9766 Local
22028 : 9767 PLOG; !INDEX INTO REMAINDER TABLE 'REM_TBL'
22029 : 9768
22030 : 9769 BAI = ONE; !FIRST LETS CLEAR THE ML11'S GOOD
22031 : 9770 IO_BUF = ZEROES; !BLOCK BY WRITING ZEROES TO IT
22032 : 9771 GO_BLK_XFER ();
22033 : 9772 MLCS1 = write;
22034 : 9773 TIME_OUT_LOOP;
22035 : 9774
22036 : 9775 incr PLOG_SEL from 0 to 6 do !TEST SYNDROME DECODE AT 7 DIFFERENT PLOG VALUES
22037 : 9776 begin
22038 : 9777
22039 : 9778 case .PLOG_SEL from 0 to 6 of !SELECT A PLOG VALUE
22040 : 9779 set
22041 : 9780
22042 : 9781 [0] : !POINTS TO CRC WORD 0
22043 : 9782 PLOG = 0;
22044 : 9783
22045 : 9784 [1] : !POINTS TO CRC WORD 1
22046 : 9785 PLOG = 1;
22047 : 9786

```



```

22049 :ML4
22050 :
22051 :
22052 : 9787 [2] : !POINTS TO CRC WORD 2
22053 : 9788 PLOG = 2;
22054 : 9789
22055 : 9790 [3] : !POINTS TO CRC WORD 3
22056 : 9791 PLOG = 3;
22057 : 9792
22058 : 9793 [4] : !POINTS TO CRC WORD 4
22059 : 9794 PLOG = 4;
22060 : 9795
22061 : 9796 [5] : !POINTS TO CRC WORD 5
22062 : 9797 PLOG = 5;
22063 : 9798
22064 : 9799 [6] : !POINTS TO ALL 6 WORDS IN CRC GROUP
22065 : 9800 PLOG = 58;
22066 : 9801 tes;
22067 : 9802
22068 : 9803 incr CHANNEL from 1 to 35 do !TEST 35 CHANNELS STARTING AT CHANNEL 1
22069 : 9804 begin
22070 : 9805 BGN SUB; !START OF SCOPE LOOP
22071 : 9806 CLR_MBUS; !CLEAR THE DRIVE
22072 : 9807
22073 : 9808 incr COUNT from 0 to 255 do !CLEAR FIRST CRC GROUP IN THE IO_BUF
22074 : 9809 IO_BUF [.COUNT] = ZEROES;
22075 : 9810
22076 : 9811 FORCE_REM (.PLOG, .CHANNEL); !FORCE ERR AT THIS WORD AND CHANNEL
22077 : 9812 CRC A = .CRC B; !CRC A EQL TO CRC B CAUSES THE UNC ERROR
22078 : 9813 GD_BLK_XFER ?; !SET UP A GOOD BLOCK TRANSFER
22079 : 9814 ECC_DIS = ZERO; !CLEAR ECC DISABLE
22080 : 9815 MLCS1 = read; !DO A READ TRANSFER
22081 : 9816 TIME_OUT_LOOP; !WAIT UNTIL THE TRANSFER IS COMPLETE
22082 : 9817
22083 : 9818 if not ((.ECH_ERR) and (.UNC_ERR)) !SEE IF THE ERROR WAS DETECTED
22084 : 9819 then
22085 : 9820 begin !REPORT ERROR IF NOT DETECTED
22086 : 9821 ERRDF (138, SYNC, DUMPER);
22087 : 9822 PRINTB (FOR_FMT, WRD_77, WRD_10, WRD_76, WRD_9);
22088 : 9823 PRINTB (FMT_21, .PLOG, .CHANNEL);
22089 : 9824 end;
22090 : 9825
22091 : 9826 incr CNT from 0 to 12 do !SEE IF ANY ERROR CORRECTION WAS DONE
22092 : 9827
22093 : 9828 if .IO_BUF [.CNT] neq ZEROES !TEST THE IO_BUF FOR ALL ZEROES
22094 : 9829 then
22095 : 9830 begin !ERROR IF ANY BITS GOT FLIPPED
22096 : 9831 ERRDF (139, SYNC, DUMPER);
22097 : 9832 PRINTB (FIV_FMT, WRD_61, WRD_74, WRD_75, WRD_12, WRD_67);
22098 : 9833 PRINTB (FMT_21, .PLOG, .CHANNEL);
22099 : 9834 end;
22100 : 9835
22101 : 9836 ENDSUB; !END OF SCOPE LOOP
22102 : 9837 end;
22103 : 9838

```

22-Dec-1980 09:24:31 TOPS-20 BLISS-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (103)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (103)

```

22105 :ML4
22106 :
22107 :
22108 :      9839      end:
22109 :      9840
22110 :      9841      ENDTST:
22114 :
22118 076464 004167 105426      $T53:      JSR      R1,$$SAVE5      :
22119 076470 152777 000010 114516      BISB     #10,$ML.REG+40      :
22120 076476 005067 113076      CLR      IO.BUF      :
22121 076502 004767 117354      JSR      PC,GD.BLK.XFER      :
22122 076506 012777 000061 114440      MOV      #61,$ML.REG      :
22123 076514 105777 114504      1$:      TSTB     $ML.REG+50      :
22124 076520 100375      BPL      1$      :
22125 076522 005005      CLR      R5      : PLOG.SEL
22126 076524 010504      2$:      MOV      R5,R4      : PLOG.SEL,*
22127 076526 006304      ASL      R4      :
22128 076530 066407 076534      ADD      3$(R4),PC      :
22129 076534 000016      3$:      .WORD   4$-3$      :
22130 076536 000022      .WORD   5$-3$      :
22131 076540 000030      .WORD   6$-3$      :
22132 076542 000036      .WORD   7$-3$      :
22133 076544 000044      .WORD   8$-3$      :
22134 076546 000052      .WORD   9$-3$      :
22135 076550 000060      .WORD  10$-3$      :
22136 076552 005003      4$:      CLR      R3      : PLOG
22137 076554 000421      BR       11$      :
22138 076556 012703 000001      5$:      MOV      #1,R3      : *,PLOG
22139 076562 000416      BR       11$      :
22140 076564 012703 000002      6$:      MOV      #2,R3      : *,PLOG
22141 076570 000413      BR       11$      :
22142 076572 012703 000003      7$:      MOV      #3,R3      : *,PLOG
22143 076576 000410      BR       11$      :
22144 076600 012703 000004      8$:      MOV      #4,R3      : *,PLOG
22145 076604 000405      BR       11$      :
22146 076606 012703 000005      9$:      MOV      #5,R3      : *,PLOG
22147 076612 000402      BR       11$      :
22148 076614 012703 000072      10$:     MOV      #72,R3      : *,PLOG
22149 076620 012704 000001      11$:     MOV      #1,R4      : *,CHANNEL
22150 076624 104402      12$:     TRAP     2      :
22151 076626 152777 000040 114360      BISB     #40,$ML.REG+40      :
22152 076634 016702 114742      MOV      ML.DUT,R2      :
22153 076640 042702 177770      BIC      #177770,R2      :
22154 076644 142777 000007 114342      BICB     #7,$ML.REG+40      :
22155 076652 150277 114336      BISB     R2,$ML.REG+40      :
22156 076656 005001      CLR      R1      : COUNT
22157 076660 010102      13$:     MOV      R1,R2      : COUNT,*
22158 076662 006302      ASL      R2      :

```

9733
9769
9770
9771
9772

9775
9778

9782
9778
9785
9776
9786
9778
9791
9778
9794
9778
9797
9778
9800
9803
9804
9805

9808
9809

Address	Hex	OpCode	Operand	Label	Comment	Seq
22160				:ML4		
22161				:		
22162						
22163	076664	005062	011600		CLR IO.BUF (R2)	
22164	076670	005201			INC R1	: COUNT 9808
22165	076672	020127	000377		CMP R1, #377	: COUNT, *
22166	076676	003770			BLE 13\$	
22167	076700	010346			MOV R3, -(SP)	: PLOG, * 9811
22168	076702	010446			MOV R4, -(SP)	: CHANNEL, *
22169	076704	004767	116672		JSR PC, FORCE.REM	
22170	076710	117702	114420		MOVB @ML, REG+1(C), R2	: 9812
22171	076714	042702	177700		BIC #177700, R2	
22172	076720	142777	000077 114376		BICB #77, @ML, REG+150	
22173	076726	150277	114372		BISB R2, @ML, REG+150	
22174	076732	004767	117124		JSR PC, GD.BLK.XFER	: 9813
22175	076736	142777	000002 114330		BICB #2, @ML, REG+120	: 9814
22176	076744	012777	000071 114202		MOV #71, @ML, REG	: 9815
22177	076752	105777	114246	14\$:	TSTB @ML, REG+50	
22178	076756	100375			BPL 14\$	
22179	076760	132777	000100 114246		BITB #100, @ML, REG+60	: 9818
22180	076766	001403			BEQ 15\$	
22181	076770	005777	114370		TST @ML, REG+210	
22182	076774	100434			BMI 16\$	
22183	076776	104455		15\$:	TRAP 55	: 9821
22184	077000	000212			.WORD 212	
22185	077002	010526			.WORD SYNC	
22186	077004	024052			.WORD DUMPER	
22187	077006	012746	006374		MOV #WRD.9, -(SP)	: 9822
22188	077012	012746	007340		MOV #WRD.76, -(SP)	
22189	077016	012746	006406		MOV #WRD.10, -(SP)	
22190	077022	012746	007346		MOV #WRD.77, -(SP)	
22191	077026	012746	006046		MOV #FOR.FMT, -(SP)	
22192	077032	012746	000005		MOV #5, -(SP)	
22193	077036	010600			MOV SP, R0	: SP, *
22194	077040	104414			TRAP 14	
22195	077042	010416			MOV R4, (SP)	: CHANNEL, * 9823
22196	077044	010346			MOV R3, -(SP)	: PLOG, *
22197	077046	012746	005534		MOV #FMT.21, -(SP)	
22198	077052	012746	000003		MOV #3, -(SP)	
22199	077056	010600			MOV SP, R0	: SP, *
22200	077060	104414			TRAP 14	
22201	077062	062706	000022		ADD #22, SP	: 9820
22202	077066	005001		16\$:	CLR R1	: CNT 9826
22203	077070	010102		17\$:	MOV R1, R2	: CNT, * 9828
22204	077072	006302			ASL R2	
22205	077074	005762	011600		TST IO.BUF (R2)	
22206	077100	001436			BEQ 18\$	
22207	077102	104455			TRAP 55	: 9831
22208	077104	000213			.WORD 213	
22209	077106	010526			.WORD SYNC	
22210	077110	024052			.WORD DUMPER	
22211	077112	012746	007240		MOV #WRD.67, -(SP)	: 9832
22212	077116	012746	006426		MOV #WRD.12, -(SP)	
22213	077122	012746	007330		MOV #WRD.75, -(SP)	
22214	077126	012746	007316		MOV #WRD.74, -(SP)	

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```

22216      :ML4
22217      :
22218
22219 077132 012746 007174      MOV      #WRD.61,-(SP)
22220 077136 012746 006062      MOV      #FIV.FMT,-(SP)
22221 077142 012746 000006      MOV      #6,-(SP)
22222 077146 010600              MOV      SP,R0          ; SP,*
22223 077150 104414              TRAP     14
22224 077152 010416              MOV      R4,(SP)       ; CHANNEL,*
22225 077154 010346              MOV      R3,-(SP)      ; PLOG,*
22226 077156 012746 005534      MOV      #FMT.21,-(SP)
22227 077162 012746 000003      MOV      #3,-(SP)
22228 077166 010600              MOV      SP,R0          ; SP,*
22229 077170 104414              TRAP     14
22230 077172 062706 000024      ADD      #24,SP
22231 077176 005201 18$:      INC      R1              ;
22232 077200 020127 000014      CMP      R1,#14        ; CNT
22233 077204 003731              BLE      17$            ; CNT,*
22234 077206 022626              CMP      (SP)+,(SP)+   ;
22235 077210 104467              TRAP     67            ;
22236 077212 006000              ROR      R0              ;
22237 077214 103603              BLO     12$            ;
22238 077216 005204              INC      R4              ; CHANNEL
22239 077220 020427 000043      CMP      R4,#43        ; CHANNEL,*
22240 077224 003002              BGT     19$
22241 077226 000167 177372      JMP     12$
22242 077232 005205 19$:      INC      R5              ; PLOG.SEL
22243 077234 020527 000006      CMP      R5,#6         ; PLOG.SEL,*
22244 077240 000102              BGT     20$
22245 077242 000167 177256      JMP     2$
22246 077246 000207 20$:      RTS      PC              ;
22247
22248      ; Routine Size: 186 words
22249      ; Maximum stack depth per invocation: 18 words
22254
22255
22259
22263 077250 177210 T53::
22264 077250 004767 1$:      JSR     PC,$T53        ;
22265 077254 104466              TRAP     66
22266 077256 006000              ROR      R0
22267 077260 103773              BLO     1$
22268 077262 000207              RTS      PC
22269

```

22271
22272
22273
22274
22275
22280
22281
22282 :

:ML4
:

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

9842 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (104)

```

22284 :ML4
22285 :
22286 :
22287 : 9843 !
22288 : 9844 BGNTST:
22289 : 9845
22290 : 9846 !++
22291 : 9847 TEST NUMBER: TST 54
22292 : 9848
22293 : 9849 TEST NAME: SINGLE BIT ERROR SYNDROME GENERATION & DECODE TEST
22294 : 9850
22295 : 9851 TEST DESCRIPTION:
22296 : 9852 TEST SYNDROME GENERATION, SYNDROME DECODE
22297 : 9853 AND ERROR CORRECTION FOR SINGLE BIT CHANNEL
22298 : 9854 ERRORS BY:
22299 : 9855
22300 : 9856 1. CLEAR THE FIRST CRC GROUP IN THE
22301 : 9857 IO_BUF AND GOOD BLOCK TO ZEROES
22302 : 9858
22303 : 9859 2. VIA DATA DIAGNOSTIC MODE FLIP A BIT
22304 : 9860 TO A ONE
22305 : 9861
22306 : 9862 3. THEN DO A MASS BUS READ TRANSFER AND
22307 : 9863 EXAMIN THE IO_BUF FOR ZEROES INDICATING
22308 : 9864 THE CORRECTION WAS PERFORMED
22309 : 9865
22310 : 9866 4. REPEAT WITH ALL BITS IN THE CRC GROUP
22311 : 9867
22312 : 9868 IMPLICIT INPUTS:
22313 : 9869 IO_BUF
22314 : 9870 A VECTOR OF 256 WORDS WHERE DATA
22315 : 9871 FOR MBUS READS AND WRITES TRANSFERS
22316 : 9872 IS FOUND.
22317 : 9873
22318 : 9874 PD_TEMP
22319 : 9875 A BIT VECTOR OF 16 BITS WHERE THE READ
22320 : 9876 PROM DATA IS STORED AND ACCESSED FROM.
22321 : 9877
22322 : 9878
22323 : 9879
22324 : 9880
22325 : 9881 local
22326 : 9882 DONE,
22327 : 9883 NIB_PAT : bitvector [4],
22328 : 9884 NIB_SEL,
22329 : 9885 GD_WRD_CNT;
22330 : 9886
22331 : 9887 incr WRD_CNT from 0 to 5 do
22332 : 9888 begin
22333 : 9889
22334 : 9890 incr BIT_CNT from 0 to 35 do
22335 : 9891 begin
22336 : 9892
22337 : 9893 incr CNT from 0 to 255 do
22338 : 9894 IO_BUF [.CNT] = ZEROES;

```

```

!DONE FLAG
!STORAGE FOR SBE DATA GENERATED
!POINTER WHERE SBE IS TO BE WRITTEN
!COUNT OF WHERE GOOD 'NIB_SEL' ARE FOUND
!FORCE SINGLE BIT ERRORS IN ONE CRC GROUP
!FORCE SBE'S AT EACH BIT OF A WORD
!CLEAR FIRST CRC GROUP OF IO_BUF

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (104)

```

22340 :ML4
22341 :
22342 :
22343 : 9895
22344 : 9896 CLR MBUS;
22345 : 9897 GD_BLK_XFER ();
22346 : 9898 MLC51 = write;
22347 : 9899 TIME_OUT_LOOP;
22348 : 9900 CLR MBUS;
22349 : 9901 DONE = ZERO;
22350 : 9902 GD_WRD_CNT = -1;
22351 : 9903 MLD1 = ZEROES;
22352 : 9904 MLD2 = ZEROES;
22353 : 9905 MLE2 = %0'060000';
22354 : 9906 NIB_SEL = .BIT_CNT/4;
22355 : 9907 DAT_LM_XFER ();
22356 : 9908 MLC51 = write;
22357 : 9909 DELAY (ONE_US);
22358 : 9910
22359 : 9911 do
22360 : 9912   begin
22361 : 9913   PD_TEMP = .MLPD;
22362 : 9914
22363 : 9915   if .PD_TEMP [.NIB_SEL] IS_SET
22364 : 9916   then
22365 : 9917     DAT_CLK = ONE
22366 : 9918   else
22367 : 9919     begin
22368 : 9920     GD_WRD_CNT = .GD_WRD_CNT + 1;
22369 : 9921
22370 : 9922
22371 : 9923     if .GD_WRD_CNT eql .WRD_CNT then DONE = ONE else DAT_CLK = ONE;
22372 : 9924
22373 : 9925     end;
22374 : 9926
22375 : 9927   end
22376 : 9928 until .DONE IS_SET;
22377 : 9929
22378 : 9930 NIB_PAT = ZEROES;
22379 : 9931 NIB_PAT [.BIT_CNT mod 4] = ONE;
22380 : 9932 D1_TEMP = ZEROES;
22381 : 9933 D2_TEMP = ZEROES;
22382 : 9934 E2_TEMP = %0'060000';
22383 : 9935 LD_LNG_WRD (.NIB_SEL, .NIB_PAT);
22384 : 9936 WRT_LNG_WRD;
22385 : 9937 DAT_CLK = ONE;
22386 : 9938 DELAY (ONE_US);
22387 : 9939 BGNSUB;
22388 : 9940 CLR MBUS;
22389 : 9941 GD_BLK_XFER ();
22390 : 9942 ECC_DIS = ZERO;
22391 : 9943 MLC51 = read;
22392 : 9944 TIME_OUT_LOOP;
22393 : 9945
22394 : 9946   incr CNT from 0 to 13 do

```

```

!CLEAR THE DRIVE
!SET UP A GOOD BLOCK TRANSFER
!DO A WRITE TRANSFER
!WAIT UNTIL THE TRANSFER IS COMPLETE
!CLEAR THE DRIVE AGAIN
!CLEAR THE DONE FLAG
!RESET THE COUNT
!LOAD THE DATA DIAG REG (BITS 0-38)
!WITH ZEROES PAT AND CRC BIT FOR
!ZEROES PAT
!CALCULATE THE NIBBLE WHERE THE SBE IS TO GO
!SET UP A DATA DIAG MODE TRANSFER
!DO A WRITE TRANSFER
!GIVE THE PROM DATA TIME TO GET OUT

!FIND THE GOOD NIB AT THE TESTED WORD

!READ THE PROM DATA FOR THIS ARRAY WORD

!IS THIS A GOOD NIBBLE

!IF NOT THEN GET THE NEXT ARRAY WORD

!ELSE SEE IF WE ARE AT THE TESTED WORD
!UP THE COUNT
!AND SEE IF WE ARE DONE

!REPEAT UNTIL WE ARE DONE

!CLEAR THE SAVE LOCATION
!AND SET THE SBE IN THIS NIBBLE
!CLEAR THE OTHER NIBBLES IN THIS ARRAY WORD
!TO BE ZEROES PATTERN AND ZEROES CRC PATTERN

!LOAD THE SBE INTO THE SAVE LOCATION
!AND WRITE THE SBE INTO THE DATA DIAG RES'S
!CLOCK THE SBE INTO THE ARRAYS MEMORY
!GIVE IT TIME TO WRITE INTO THE MEMORY
!START OF SCOPE LOOP
!CLEAR OUT THE DATA DIAG MODE XFERR
!SET UP A GOOD BLOCK TRANSFER
!CLEAR ECC DISABLE
!AND READ OUT THE SBE

!SEE IF THE SBE WAS CORRECTED

```

22-Dec-1980 09:24:31 TOPS-20 Bl'ss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (104)

```

22396 :ML4
22397 :
22398 :
22399 : 9947
22400 : 9948      if .IO_BUF [.CNT] neq ZEROES      !TEST THE IO_BUF FOR ALL ZEROES
22401 : 9949      then
22402 : 9950          begin                      !ERROR IF THE ERROR WAS NOT CORRECTED
22403 : 9951          ERRDF (140, SYNC, DUMPER);
22404 : 9952          PRINTB (THR_FMT, PHR_12, WRD_76, WRD_9);
22405 : 9953          PRINTB (FMT_22, .WRD_CNT, .BIT_CNT);
22406 : 9954          end;
22407 : 9955
22408 : 9956      ENDSUB;                          !END OF SCOPE LOOP
22409 : 9957      end;
22410 : 9958
22411 : 9959      end;
22412 : 9960
22413 : 9961      ENDTST;
22417 :

```

```

22421 077264 004167 104626      $T54:   JSR      R1, $SAVES                    :           9841
22422 077270 162706 000010      SUB      #10, SP                          :
22423 077274 005001              CLR      R1                               : WRD.CNT   9887
22424 077276 005002      1$:    CLR      R2                       : BIT.CNT   9890
22425 077300 005003      2$:    CLR      R3                       : CNT       9893
22426 077302 010304      3$:    MOV      R3, R4                    : CNT,*    9894
22427 077304 006304              ASL      R4
22428 077306 005064 011600      CLR      IO_BUF (R4)
22429 077312 005203              INC      R3                               : CNT       9893
22430 077314 020327 000377      CMP      R3, #377                         : CNT,*
22431 077320 003770              BLE      3$
22432 077322 152777 000040 113664  BISB     #40, @ML.REG+40                  :           9894
22433 077330 016704 114246      MOV      ML.DUT, R4
22434 077334 042704 177770      BIC      #177770, R4
22435 077340 142777 000007 113646  BICB     #7, @ML.REG+40
22436 077346 150477 113642      BISB     R4, @ML.REG+40
22437 077352 004767 116504      JSR      PC, GD.BLK.XFER                  :           9897
22438 077356 012777 000061 113570  MOV      #61, @ML.REG                    :           9898
22439 077364 105777 113634      4$:    TSTB     @ML.REG+50
22440 077370 100375              BPL      4$
22441 077372 152777 000040 113614  BISB     #40, @ML.REG+40                  :           9899
22442 077400 016704 1141 6      MOV      ML.DUT, R4
22443 077404 042704 177770      BIC      #177770, R4
22444 077410 142777 000007 113576  BICB     #7, @ML.REG+40
22445 077416 150477 113572      BISB     R4, @ML.REG+40
22446 077422 005066 000002      CLR      2(SP)                          : DONE     9901
22447 077426 012716 177777      MOV      #-1, (SP)                       : *,GD.WRD.CNT 9902
22448 077432 005077 113706      CLR      @ML.REG+170                      :           9903
22449 077436 005077 113712      CLR      @ML.REG+200                     :           9904

```


Address	Op-Code	Operand 1	Operand 2	Operand 3	Instruction	Comments	Timestamp	Page
22451						:ML4	22-Dec-1980 09:24:31	TOPS
22452						:	22-Dec-1980 09:21:22	PA:<
22453								
22454	077442	012777	060000	113664	MOV #60000,@ML.REG+160	:		9905
22455	077450	010246			MOV R2,-(SP)	: BIT.CNT,*		9906
22456	077452	012746	000004		MOV #4,-(SP)			
22457	077456	004767	104270		JSR PC,BL\$DIV			
22458	077462	010005			MOV R0,R5	: *,NIB.SEL		
22459	077464	004767	114472		JSR PC,DAT.DM.XFER	:		9907
22460	077470	012777	000061	113456	MOV #61,@ML.REG	:		9908
22461	077476	012703	000001		MOV #1,R3	: *,SSTMP2		9909
22462	077502	001411		5\$:	BEQ 8\$			
22463	077504	016704	102406		MOV L\$DLY,R4	: *,SSTMP1		
22464	077510	001404			BEQ 7\$			
22465	077512	005066	000012	6\$:	CLR 12(SP)	: SSTMP		
22466	077516	005304			DEC R4	: SSTMP1		
22467	077520	001374			BNE 6\$			
22468	077522	005303		7\$:	DEC R3	: SSTMP2		
22469	077524	000766			BR 5\$			
22470	077526	010504		8\$:	MOV R5,R4	: NIB.SEL,*		9915
22471	077530	006204			ASR R4			
22472	077532	006204			ASR R4			
22473	077534	006204			ASR R4			
22474	077536	062704	013120		ADD #PD.TEMP,R4			
22475	077542	017767	113636	113350	MOV @ML.REG+230,PD.TEMP	:		9913
22476	077550	010446			MOV R4,-(SP)	:		9915
22477	077552	010546			MOV R5,-(SP)	: NIB.SEL,*		
22478	077554	042716	177770		BIC #177770,(SP)			
22479	077560	012746	000001		MOV #1,-(SP)			
22480	077564	005046			CLR -(SP)			
22481	077566	004767	103346		JSR PC,BL\$GT2			
22482	077572	062706	000010		ADD #10,SP			
22483	077576	005300			DEC R0			
22484	077600	001411			BEQ 10\$:		9917
22485	077602	005266	000004		INC 4(SP)	: GD.WRD.CNT		9920
22486	077606	026601	000004		CMP 4(SP),R1	: GD.WRD.CNT,WRD.CNT		9923
22487	077612	001004			BNE 10\$			
22488	077614	012766	000001	000006	MOV #1,6(SP)	: *,DONE		
22489	077622	000403			BR 11\$			
22490	077624	152777	000020	113442	BISB #20,@ML.REG+120			
22491	077632	026627	000006	000001	CMP 6(SP),#1	: DONE,*		9928
22492	077640	001340			BNE 9\$			
22493	077642	005066	000010		CLR 10(SP)	: NIB.PAT		9930
22494	077646	010246			MOV R2,-(SP)	: BIT.CNT,*		9931
22495	077650	012746	000004		MOV #4,-(SP)			
22496	077654	004767	104104		JSR PC,BL\$MOD			
22497	077660	010004			MOV R0,R4			
22498	077662	006200			ASR R0			
22499	077664	006200			ASR R0			
22500	077666	006200			ASR R0			
22501	077670	012703	000014		MOV #14,R3			
22502	077674	060603			ADD SP,R3	: NIB.PAT,*		
22503	077676	060300			ADD R3,R0			
22504	077700	010016			MOV R0,(SP)			
22505	077702	010446			MOV R4,-(SP)			

22507										22-Dec-1980 09:24:31	TOPS
22508										22-Dec-1980 09:21:22	PA:<
22509											
22510	077704	042716	177770		BIC	#177770, (SP)					
22511	077710	012746	000001		MOV	#1, -(SP)					
22512	077714	011646			MOV	(SP), -(SP)					
22513	077716	004767	103454		JSR	PC, BL\$PU2					
22514	077722	005067	111224		CLR	D1.TEMP	:				9932
22515	077726	005067	111222		CLR	D2.TEMP	:				9933
22516	077732	012767	060000	111216	MOV	#60000, E2.TEMP	:				9934
22517	077740	010516			MOV	R5, (SP)	:				9935
22518	077742	016646	000022		MOV	22(SP), -(SP)	:	NIB.SEL, *			
22519	077746	004767	117152		JSR	PC.LD.LNG.WRD	:	NIB.PAT, *			
22520	077752	016777	111174	113364	MOV	D1.TEMP, @ML.REG+170					
22521	077760	016777	111170	113366	MOV	D2.TEMP, @ML.REG+200					
22522	077766	016777	111164	113340	MOV	E2.TEMP, @ML.REG+160					
22523	077774	152777	000020	113272	BISB	#20, @ML.REG+120	:				9937
22524	100002	012703	000001		MOV	#1, R3	:	*.SSTMP2			9938
22525	100006	001411		12\$:	BEQ	15\$:				
22526	100010	016704	102102		MOV	LS\$DLY, R4	:	*.SSTMP1			
22527	100014	001404			BEQ	14\$:				
22528	100016	005066	000026	13\$:	CLR	26(SP)	:	SSTMP			
22529	100022	005304			DEC	R4	:	SSTMP1			
22530	100024	001374			BNE	13\$:				
22531	100026	005303		14\$:	DEC	R3	:	SSTMP2			
22532	100030	000766			BR	12\$:				
22533	100032	104402		15\$:	TRAP	2	:				
22534	100034	152777	000040	113152	BISB	#40, @ML.REG+40	:				9939
22535	100042	016704	113534		MOV	ML.DUT, R4					
22536	100046	042704	177770		BIC	#177770, R4					
22537	100052	142777	000007	113134	BICB	#7, @ML.REG+40					
22538	100060	150477	113130		BISB	R4, @ML.REG+40					
22539	100064	004767	115772		JSR	PC, ED.BLK.XFER	:				9941
22540	100070	142777	000002	113176	BICB	#2, @ML.REG+120	:				9942
22541	100076	012777	000071	113050	MOV	#71, @ML.REG	:				9943
22542	100104	105777	113114	16\$:	TSTB	@ML.REG+50	:				
22543	100110	100375			BPL	16\$:				
22544	100112	005003			CLR	R3	:	CNT			9946
22545	100114	010304		17\$:	MOV	R3, R4	:	CNT, *			9948
22546	100116	006304			ASL	R4					
22547	100120	005764	011600		TST	10.BUF(R4)					
22548	100124	001432			BEQ	18\$					
22549	100126	104455			TRAP	55	:				9951
22550	100130	000214			.WORD	214					
22551	100132	010526			.WORD	SYNC					
22552	100134	024052			.WORD	DUMPER					
22553	100136	012746	006374		MOV	#WRD.9, -(SP)	:				9952
22554	100142	012746	007340		MOV	#WRD.76, -(SP)					
22555	100146	012746	007614		MOV	#PHR.12, -(SP)					
22556	100152	012746	006034		MOV	#THR.FMT, -(SP)					
22557	100156	012746	000004		MOV	#4, -(SP)					
22558	100162	010600			MOV	SP, R0	:	SP, *			
22559	100164	104414			TRAP	14					
22560	100166	010216			MOV	R2, (SP)	:	BIT.CNT, *			9953
22561	100170	010146			MOV	R1, -(SP)	:	WRD.CNT, *			

```

22563      :ML4
22564      :
22565
22566 100172 012746 005606      MOV      #FMT.22, -(SP)
22567 100176 012746 000003      MOV      #3, -(SP)
22568 100202 010600              MOV      SP, R0          ; SP,*
22569 100204 104414              TRAP     14
22570 100206 062706 000020      ADD      #20, SP
22571 100212 005203 18$:      INC      R3              ; CNT
22572 100214 020327 000015      CMP      R3, #15        ; CNT,*
22573 100220 003735              BLE      17$
22574 100222 104467              TRAP     67              ;
22575 100224 006000              ROR      R0
22576 100226 103701              BLO     15$
22577 100230 062706 000020      ADD      #20, SP
22578 100234 005202              INC      R2              ; BIT.CNT
22579 100236 020227 000043      CMP      R2, #43        ; BIT.CNT,*
22580 100242 003002              BGT     19$
22581 100244 000167 177030      JMP      2$
22582 100250 005201 19$:      INC      R1              ; WRD.CNT
22583 100252 020127 000005      CMP      R1, #5         ; WRD.CNT,*
22584 100256 003002              BGT     20$
22585 100260 000167 177012      JMP      1$
22586 100264 062706 000010 20$:      ADD      #10, SP
22587 100270 000207              RTS      PC              ;
22588
22589      : Routine Size: 259 words
22590      : Maximum stack depth per invocation: 26 words
22595
22596
22600
22604 100272
22605 100272 004767 176766  T54::
22606 100276 104466 1$:      JSR      PC, $T54
22607 100300 006000              TRAP     66              ;
22608 100302 103773              ROR      R0
22609 100304 000207              BLO     1$
22610
22611      : Routine Size: 6 words
22612      : Maximum stack depth per invocation: 0 words

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

9950
9946

9954

9891
9890

9887

9841

9959

CZMLABO ML-11 LOGIC TEST
MISCELLANEOUS CODING SECTION

MACRO M1113 22-DEC-80 12:16 PAGE 486^{F 5}

SEQ 0473

22618 :ML4
22619 :
22620 :
22621 :

9962 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (104)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (105)

```

22623 :ML4
22624 :
22625 :
22626 : 9963 !
22627 : 9964 ! BGNTST:
22628 : 9965
22629 : 9966 ! ++
22630 : 9967 ! TEST NUMBER: TST 55
22631 : 9968
22632 : 9969 ! TEST NAME: MULTIPLE BIT ERROR SYNDROME GENERATION AND DECODE TEST
22633 : 9970
22634 : 9971 ! TEST DESCRIPTION:
22635 : 9972 ! TEST SYNDROME GENERATION, SYNDROME DECODE
22636 : 9973 ! AND ERROR CORRECTION FOR MULTIPLE BIT
22637 : 9974 ! CHANNEL ERRORS BY:
22638 : 9975
22639 : 9976 ! 1. CLEAR THE FIRST CRC GROUP IN THE
22640 : 9977 ! IO_BUF AND GOOD BLOCK TO ZEROES
22641 : 9978
22642 : 9979 ! 2. VIA DATA DIAG MODE FLIP ALL BITS IN
22643 : 9980 ! A CHANNEL TO ONES
22644 : 9981
22645 : 9982 ! 3. THEN DO A MASS BUS READ TRANSFER AND
22646 : 9983 ! EXAMIN THE IO BUF FOR ZEROES INDICATING
22647 : 9984 ! THE CORRECTION WAS PERFORMED
22648 : 9985
22649 : 9986 ! 4. REPEAT FOR ALL CHANNELS IN THE CRC GROUP
22650 : 9987
22651 : 9988 ! IMPLICIT INPUTS:
22652 : 9989 ! PD TEMP
22653 : 9990 ! A BIT VECTOR OF 16 BITS WHERE THE READ
22654 : 9991 ! FROM DATA IS STORED OAND ACCESSED FROM.
22655 : 9992
22656 : 9993 ! IO_BUF
22657 : 9994 ! A VECTOR OF 256 WORDS WHERE DATA
22658 : 9995 ! FOR MBUS READS AND WRITES TRANSFERS
22659 : 9996 ! IS FOUND.
22660 : 9997
22661 : 9998
22662 : 9999 !
22663 : 10000
22664 : 10001 local
22665 : 10002 NIB_PAT : bitvector [4],
22666 : 10003 NIB_SEL,
22667 : 10004 GD_WRD_CNT;
22668 : 10005
22669 : 10006 E2_TEMP<12, 3> = %b'110';
22670 : 10007
22671 : 10008 incr BIT_CNT from 0 to 35 do
22672 : 10009 begin
22673 : 10010 CLR_MBUS;
22674 : 10011
22675 : 10012 incr CNT from 0 to 255 do
22676 : 10013 IO_BUF [.CNT] = ZEROES;
22677 : 10014

```

```

! STORAGE FOR SBE TO BE GENERATED
! POINTER TO WHERE ERROR IS TO BE WRITTEN
! COUNT OF WHERE GOOD 'NIB_SEL' ARE FOUND

! LOAD DATA STRUCTURE WITH ZEROES CRC PATTERN

! TEST FOR MULTILPLE ERRORS AT 36 CHANNELS

! CLEAR THE DRIVE

! CLEAR FIRST CRC GROUP IN THE IO_BUF

```

```

22679 :ML4
22680 :
22681 :
22682 :      10015 GD_BLK_XFER ();
22683 :      10016 MLCS1 = write;
22684 :      10017 TIME_OUT_LOOP;
22685 :      10018 GD_WRD_CNT = ZERO;
22686 :      10019 D1_TEMP = ZEROES;
22687 :      10020 D2_TEMP = ZEROES;
22688 :      10021 NIB_PAT = ZEROES;
22689 :      10022 NIB_SEL = .BIT_CNT/4;
22690 :      10023 NIB_PAT [.BIT_CNT mod 4] = ONE;
22691 :      10024 LD [NG WRD (.NIB_SEL, .NIB_PAT)];
22692 :      10025 WRT_LNG WRD;
22693 :      10026 DAT_DM_XFER ();
22694 :      10027 MLCS1 = write;
22695 :      10028 DELAY (ONE_US);
22696 :
22697 :      10029
22698 :      10030 do
22699 :      10031 begin
22700 :      10032 PD_TEMP = .MLPD;
22701 :      10033
22702 :      10034 if .PD_TEMP [.NIB_SEL] IS_SET
22703 :      10035 then
22704 :      10036 DAT_CLK = ONE
22705 :      10037 else
22706 :      10038 begin
22707 :      10039 GD_WRD_CNT = .GD_WRD_CNT + 1;
22708 :      10040 DAT_CLR = ONE;
22709 :      10041 end;
22710 :      10042
22711 :      10043 end
22712 :      10044 until .GD_WRD_CNT eql 6;
22713 :      10045
22714 :      10046 BGNSUB;
22715 :      10047 CLR MBUS;
22716 :      10048 GD_BLK_XFER ();
22717 :      10049 ECC_DIS = 0;
22718 :      10050 MLCS1 = read;
22719 :      10051 TIME_OUT_LOOP;
22720 :      10052
22721 :      10053 incr WRD_CNT from 0 to 13 do
22722 :      10054
22723 :      10055 if .IO_BUF [.WRD_CNT] neq ZEROES
22724 :      10056 then
22725 :      10057 begin
22726 :      10058 ERRDF (141, SYNC, DUMPER);
22727 :      10059 PRINTB (THR_FMT, PHR_13, WRD_76, WRD_9);
22728 :      10060 PRINTB (FMT_22, .WRD_CNT, .BIT_CNT);
22729 :      10061 end;
22730 :      10062
22731 :      10063 ENDSUB;
22732 :      10064 end;
22733 :      10065
22734 :      10066 ENDTST;

```

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (105)

!ALSO CLEAR THE FIRST CRC GROUP
!IN THE ML11'S GOOD BLOCK

!CLEAR COUNT
!CLEAR D1_TEMP
!CLEAR D2_TEMP
!CLEAR NIBBLE PATTERN
!CALCULATE THE NIBBLE TO BE IN ERROR
!CALCULATE AND THE BIT TO BE IN ERROR
!LOAD THE ERROR IN D1 TEMP OR D2 TEMP
!LOAD THE REGISTER WITH THE ERROR
!SET UP A DATA DIAG MODE TRANSFER
!DO A WRITE TRANSFER
!GIVE THE PROM DATA TIME TO COME OUT

!LOAD THIS CHANNEL WITH MULTIPLE ERRORS

!READ THIS ARRAY WORDS PROM_DATA

!SEE IF THIS A GOOD NIBBLE

!CLOCK OUT ANOTHER ARRAY WORD IF BAD

!ELSE CLOCK IN THE ERROR INTO A GOOD NIBBLE

!REPEAT UNTIL 6 WORDS ARE WRITTEN

!START OF SCOPE LOOP
!CLEAR OUT THE DATA DISG MODE TRANSFER
!SET UP A GOOD BLOCK TRANSFER
!ENABLE ERROR CORRECTION
!READ OUT THE ERRORS
!WAIT FOR THE TRANSFER TO COMPLETE

!SEE IF THE CORRECTIONS WERE DONE

!TEST THE IO_BUF FOR ALL ZEROES

!REPORT THE ERROR IF ANY LOCATIONS ARE NOT ZEROES

!END OF SCOPE LOOP

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (105)

```

22735 :ML4
22736 :
22737 :
22741 :
22745 100306 004167 103604      $T55: JSR    R1,$SAVE5          ;
22746 100312 024646          CMP    -(SP),-(SP)          ;
22747 100314 042767 070000 110634 BIC    #70000,E2.TEMP      ;
22748 100322 052767 060000 110626 BIS    #60000,E2.TEMP      ;
22749 100330 005003          CLR    R3                   ; BIT.CNT
22750 100332 152777 000040 112654 1$:  BISB  #40,@ML.REG+40      ;
22751 100340 016702 113236          MOV    ML,DUT,R2           ;
22752 100344 042702 177770          BIC    #177770,R2         ;
22753 100350 142777 000007 112636 BICB  #7,@ML.REG+40      ;
22754 100356 150277 112632          BISB  R2,@ML.REG+40      ;
22755 100362 005001          CLR    R1                   ; CNT
22756 100364 010102          MOV    R1,R2              ; CNT,*
22757 100366 006302          ASL   R2                   ;
22758 100370 005062 011600          CLR   IO.BUF(R2)         ;
22759 100374 005201          INC   R1                   ; CNT
22760 100376 020127 000377          CMP   R1,#377            ; CNT,*
22761 100402 003770          BLE  2$                   ;
22762 100404 004767 115452          JSR   PC,GD.BLK.XFER     ;
22763 100410 012777 000061 112536 MOV    #61,@ML.REG        ;
22764 100416 105777 112602          TSTB @ML.REG+50         ;
22765 100422 100375          BPL  3$                   ;
22766 100424 005005          CLR   R5                   ; GD.WRD.CNT
22767 100426 005067 110520          CLR   D1.TEMP            ;
22768 100432 005067 110516          CLR   D2.TEMP            ;
22769 100436 005016          CLR   (SP)               ; NIB.PAT
22770 100440 010346          MOV   R3,-(SP)           ; BIT.CNT,*
22771 100442 012746 000004          MOV   #4,-(SP)           ;
22772 100446 004767 103300          JSR   PC,BL$DIV          ;
22773 100452 010004          MOV   R0,R4              ; *,NIB.SEL
22774 100454 010346          MOV   R3,-(SP)           ; BIT.CNT,*
22775 100456 012746 000004          MOV   #4,-(SP)           ;
22776 100462 004767 103276          JSR   PC,BL$MOD          ;
22777 100466 010002          MOV   R0,R2              ;
22778 100470 006200          ASR   R0                   ;
22779 100472 006200          ASR   R0                   ;
22780 100474 006200          ASR   R0                   ;
22781 100476 012701 000010          MOV   #10,R1             ;
22782 100502 000601          ADD   SP,R1               ; NIB.PAT,*
22783 100504 060100          ADD   R1,R0               ;
22784 100506 010016          MOV   R0,(SP)            ;
22785 100510 010246          MOV   R2,-(SP)           ;
22786 100512 042716 177770          BIC   #177770,(SP)       ;
22787 100516 012746 000001          MOV   #1,-(SP)           ;
22788 100522 011646          MOV   (SP),-(SP)         ;

```

22790 :ML4
22791 :
22792 :
22793 100524 004767 102646 JSR PC,BLSPU2
22794 100530 010416 MOV R4,(SP) ; NIB.SEL,* 1002
22795 100532 016646 000016 MOV 16(SP),-(SP) ; NIB.PAT,*
22796 100536 004767 116362 JSR PC,LD.LNG.WRD
22797 100542 016777 110404 112574 MOV D1.TEMP,@ML.REG+170
22798 100550 016777 110400 112576 MOV D2.TEMP,@ML.REG+200
22799 100556 016777 110374 112550 MOV E2.TEMP,@ML.REG+160
22800 100564 004767 113372 JSR PC,DAT.DM.XFER ;
22801 100570 012777 000061 112356 MOV #61,@ML.REG ;
22802 100576 012701 000001 MOV #1,R1 ; *,SSTMP2 1002
22 03 100602 001411 4\$: BEQ 7\$; * 1002
22804 100604 016702 101306 MOV L\$DLY,R2 ; *,SSTMP1
22805 100610 001404 BEQ 6\$
22806 100612 005066 000022 5\$: CLR 22(SP) ; \$STMP
22807 100616 005302 DEC R2 ; \$STMP1
22808 100620 001374 BNE 5\$
22809 100622 005301 6\$: DEC R1 ; \$STMP2
22810 100624 000766 BR 4\$
22811 100626 010402 7\$: MOV R4,R2 ; NIB.SEL,* 1003
22812 100630 006202 ASR R2
22813 100632 006202 ASR R2
22814 100634 006202 ASR R2
22815 100636 062702 013120 ADD #PD.TEMP,R2
22816 100642 017767 112536 112250 8\$: MOV @ML.REG+230,PD.TEMP ;
22817 100650 010246 MOV R2,-(SP) ;
22818 100652 010446 MOV R4,-(SP) ; NIB.SEL,* 1003
22819 100654 042716 177770 BIC #177770,(SP)
22820 100660 012746 000001 MOV #1,-(SP)
22821 100664 005046 CLR -(SP)
22822 100666 004767 102246 JSR PC,BLSGT2
22823 100672 062706 000010 ADD #10,SP
22824 100676 005300 DEC R0
22825 100700 001401 BEQ 9\$; 1003
22826 100702 005205 INC R5 ; GD.WRD.CNT 1003
22827 100704 152777 000020 112362 9\$: BISB #20,@ML.REG+120 ; 1004
22828 100712 020527 000006 CMP R5,#6 ; GD.WRD.CNT,* 1004
22829 100716 001351 BNE 8\$
22830 100720 104402 10\$: TRAP 2
22831 100722 152777 000040 112264 BISB #40,@ML.REG+40 ; 1004
22832 100730 016702 112646 MOV ML,DUT,R2
22833 100734 042702 177770 BIC #177770,R2
22834 100740 142777 000007 112246 BICB #7,@ML.REG+40
22835 100746 150277 112242 BISB R2,@ML.REG+40
22836 100752 004767 115104 JSR PC,GD.BLK.XFER ; 1004
22837 100756 142777 000002 112310 BICB #2,@ML.REG+120 ; 1004
22838 100764 012777 000071 112162 MOV #71,@ML.REG ; 1005
22839 100772 105777 112226 11\$: TSTB @ML.REG+50
22840 100776 100375 BPL 11\$
22841 101000 005001 CLR R1 ; WRD.CNT 1005
22842 101002 010102 12\$: MOV R1,R2 ; WRD.CNT,* 1005
22843 101004 006302 ASL R2
22844 101006 005762 011600 TST IO.BUF(R2)


```

22846      :ML4
22847      :
22848
22849 101012 001432      BEQ      13$
22850 101014 104455      TRAP     55      ;
22851 101016 000215      .WORD   215
22852 101020 010526      .WORD   SYNC
22853 101022 024052      .WORD   DUMPER
22854 101024 012746 006374  MOV     #WRD.9,-(SP) ;
22855 101030 012746 007340  MOV     #WRD.76,-(SP)
22856 101034 012746 007636  MOV     #PHR.13,-(SP)
22857 101040 012746 006034  MOV     #THR.FMT,-(SP)
22858 101044 012746 000004  MOV     #4,-(SP)
22859 101050 010600      MOV     SP,R0      ; SP,*
22860 101052 104414      TRAP    14
22861 101054 010316      MOV     R3,(SP)    ; BIT.CNT,*
22862 101056 010146      MOV     R1,-(SP)   ; WRD.CNT,*
22863 101060 012746 005606  MOV     #FMT.22,-(SP)
22864 101064 012746 000003  MOV     #3,-(SP)
22865 101070 010600      MOV     SP,R0      ; SP,*
22866 101072 104414      TRAP    14
22867 101074 062706 000020  ADD     #20,SP
22868 101100 005201      13$:   INC     R1      ;
22869 101102 020127 000015  CMP     R1,#15     ; WRD.CNT
22870 101106 003735      BLE     12$        ; WRD.CNT,*
22871 101110 104467      TRAP    67      ;
22872 101112 006000      ROR     R0
22873 101114 103701      BLO     10$
22874 101116 062706 000020  ADD     #20,SP
22875 101122 005203      INC     R3      ;
22876 101124 020327 000043  CMP     R3,#43     ; BIT.CNT
22877 101130 003002      BGT     14$        ; BIT.CNT,*
22878 101132 000167 177174  JMP     1$
22879 101136 022626      14$:   CMP     (SP)+,(SP)+ ;
22880 101140 000207      RTS     PC
22881
22882      ; Routine Size: 206 words
22883      ; Maximum stack depth per invocation: 24 words
22888
22889
22893
22897 101142      T55::
22898 101142 004767 177140  1$:   JSR     PC,$T55
22899 101146 104466      TRAP    66      ;

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

1005

1005

1006

1005

1005

1006

1000

1000

9961

1006

22901
22902
22903
22904 101150 006000
22905 101152 103773
22906 101154 000207
22907
22908
22909
22914
22915
22916 :

:ML4
:

ROR R0
BLO 1\$
RTS PC

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

10067 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

22918 :ML4

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (106)

```

22921 : 10068 !
22922 : 10069 BGNTST;
22923 : 10070
22924 : 10071 !++
22925 : 10072 TEST NUMBER: TST 56
22926 : 10073
22927 : 10074 TEST NAME: ECC ERROR REGISTER TEST
22928 : 10075
22929 : 10076 TEST DESCRIPTION:
22930 : 10077 TEST THE ECC ERROR REGISTER FOR
22931 : 10078 CLEARING AND LATCHING OF ECC ERROR
22932 : 10079 INFORMATION ON DETECTION
22933 : 10080 OF ECC ERRORS BY:
22934 : 10081
22935 : 10082 1. THIS TEST IS TABLE DRIVEN. DATA TABLE (DT_1)
22936 : 10083 CONTAINS ERROR FORCING DATA AND EXPECTED
22937 : 10084 MLEE DATA AS A RESULT OF THE FORCED ERROR
22938 : 10085
22939 : 10086 2. THE TEST THEREFOR INDEXES INTO DT_1 TO FORCE
22940 : 10087 AN ECC ERROR
22941 : 10088
22942 : 10089 3. A MASS BUS READ TRANSFER IS DONE
22943 : 10090
22944 : 10091 4. DT_1 IS AGAIN INDEXED AND ITS CONTENTS
22945 : 10092 IS COMPARED AGAINST THE RESULTING
22946 : 10093 MLEE REGISTER CONTENTS
22947 : 10094
22948 : 10095 5. A MASS BUS CLEAR IS DONE AN MLEE IS CHECKED FOR
22949 : 10096 ZEROES
22950 : 10097
22951 : 10098 6. THE ECH BIT IS ALSO EXAMINED FOR CORRECT
22952 : 10099 LATCHING INFORMATION
22953 : 10100
22954 : 10101
22955 : 10102
22956 : 10103 --
22957 : 10104
22958 : 10105 local
22959 : 10106 FINISH, !VARIABLE ENDING CONDITION FOR LATCH_LOOP
22960 : 10107 ERR_FLG; !ERROR FLAG
22961 : 10108
22962 : 10109 CLR MBUS; !CLEAR THE DRIVE
22963 : 10110 FINISH = 4; !MAKE LATCH_LOOP DO 5 LOOPS AT FIRST
22964 : 10111
22965 : 10112 incr CLR_LOOP from 0 to 1 do !THE FIRST PASS TESTS MLEE FOR LATCHING
22966 : 10113 !THE SECOND PASS TESTS MLEE FOR CLEARING
22967 : 10114 begin
22968 : 10115
22969 : 10116 incr LATCH_LOOP from 0 to .FINISH do !TEST MLEE FOR LATCHING BY USING DT_1 DATA
22970 : 10117 begin
22971 : 10118 BGNSUB; !START OF SCOPE LOOP
22972 : 10119 ERR_FLG = ZERO; !CLEAR THE ERROR FLAG

```

22974 :ML4

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (106)

```

22977 : 10120 MLEER = ZEROES; !CLEAR THE ERROR REGISTER
22978 : 10121 ECC_DM = ONE; !ENABLE ECC DIAG MODE
22979 : 10122 PAR_CRC_WRD = .DT_1 [.LATCH_LOOP, PO_5]; !GET PAR CRC WRD DATA FROM DT_1
22980 : 10123 CRC_A = .DT_1 [.LATCH_LOOP, AO_5]; !GET CRC_A FROM DT_1
22981 : 10124 CRC_B = .DT_1 [.LATCH_LOOP, BO_5]; !GET CRC_B FROM DT_1
22982 : 10125 GD_BLK_XFER(); !SET Lo A GOOD BLOCK TRANSFER
22983 : 10126 ECC_DIS = ZERO; !DISABLE ECC DIABLE
22984 : 10127 MLC51 = read; !READ THE ERROR AND LACTH MLEE
22985 : 10128 TIME_OUT_LOOP;
22986 : 10129
22987 : 10130 if .CRC_ERR neq .DT_1 [.LATCH_LOOP, CRC_DATA] then ERR_FLG = ONE; !TEST CRC_ERR
22988 : 10131
22989 : 10132 if .SGL_ERR neq .DT_1 [.LATCH_LOOP, SGL_DATA] then ERR_FLG = ONE; !TEST SGL_ERR
22990 : 10133
22991 : 10134 if .UNC_ERR neq .DT_1 [.LATCH_LOOP, UNC_DATA] then ERR_FLG = ONE; !TEST UNC
22992 : 10135
22993 : 10136 if .BIT_IN_ERR neq .DT_1 [.LATCH_LOOP, EO_5] then ERR_FLG = ONE; !TEST BIT IN ERROR
22994 : 10137
22995 : 10138 if .CHAN_IN_ERR neq .DT_1 [.LATCH_LOOP, CO_5] then ERR_FLG = ONE; !TEST CHANNEL IN ERROR
22996 : 10139
22997 : 10140 if .ECH_ERR neq .DT_1 [.LATCH_LOOP, ECH_DATA] !TEST IF ECH GOT SET
22998 : 10141 then
22999 : 10142 begin !REPORT ERROR IF NOT SET
23000 : 10143 ERRDF (123, ASYNC, DUMPER);
23001 : 10144 PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_69, WRD_14);
23002 : 10145 end;
23003 : 10146
23004 : 10147 if .ERR_FLG IS_SET !TEST IF THE ERROR FLAG GOT SET
23005 : 10148 then
23006 : 10149 begin !REPORT ERROR IF SET
23007 : 10150 ERRDF (124, SYNC, DUMPER);
23008 : 10151 PRINTB (THR_FMT, REG_14, WRD_52, PHR_11);
23009 : 10152 PRINTB (FMT_1, .DT_1 [.LATCH_LOOP, EE_DATA], .MLEE);
23010 : 10153 PRINTB (FMT_19, .DT_1 [.LATCH_LOOP, BO_5], .DT_1 [.LATCH_LOOP, AO_5], .DT_1 [.LATCH_LOOP, PO_5])
23011 : 10154 end;
23012 : 10155
23013 : 10156 ENDSUB; !END OF SCOPE LOOP
23014 : 10157 end;
23015 : 10158
23016 : 10159 CLR_MBUS; !CLEAR THE REGISTER
23017 : 10160
23018 : 10161 if .MLEE neq ZERO !SEE IF THE REGISTER GOT CLEARED
23019 : 10162 then
23020 : 10163 begin !REPORT ERROR IF NOT CLEARED
23021 : 10164 ERRDF (125, SYNC, DUMPER);
23022 : 10165 PRINTB (FOR_FMT, REG_14, WRD_52, FNC_7, WRD_14);
23023 : 10166 end;
23024 : 10167
23025 : 10168 ECC_DM = ONE; !ENABLE ECC DISABLE
23026 : 10169 FINISH = ZERO; !MAKE LATCH_LOOP LOOP ONCE TO CLEAR MLEE AGAIN
23027 : 10170 end;
23028 : 10171

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (106)

23030 :ML4
23031 :
23032 :
23033 :

10172 ENDTST:

23041	101156	004167	102734		\$T56:	JSR	R1,\$SAVE5	:	1006
23042	101162	152777	000040	112024		BISB	#40,@ML.REG+40	:	1010
23043	101170	016705	112406			MOV	ML,DUT,R5	:	
23044	101174	042705	177770			BIC	#177770,R5	:	
23045	101200	142777	000007	112006		BICB	#7,@ML.REG+40	:	
23046	101206	150577	112002			BISB	R5,@ML.REG+40	:	
23047	101212	012746	000004			MOV	#4,-(SP)	:	1011
23048	101216	005046				CLR	-(SP)	:	1011
23049	101220	005004			1\$:	CLR	R4	:	1011
23050	101222	000167	000644			JMP	16\$:	1011
23051	101226	010446			2\$:	MOV	R4,-(SP)	:	1012
23052	101230	012746	000006			MOV	#6,-(SP)	:	
23053	101234	004767	102266			JSR	PC,BL\$MUL	:	
23054	101240	010005				MOV	R0,R5	:	
23055	101242	012701	013534			MOV	#DT.1,R1	:	1013
23056	101246	060501				ADD	R5,R1	:	
23057	101250	012702	013536			MOV	#DT.1+2,R2	:	1012
23058	101254	060502				ADD	R5,R2	:	
23059	101256	104402			3\$:	TRAP	2	:	1011
23060	101260	005003				CLR	R3	:	1011
23061	101262	005077	111746			CLR	@ML.REG+60	:	1012
23062	101266	152777	000001	112000		BISB	#1,@ML.REG+120	:	1012
23063	101274	011200				MOV	(R2),R0	:	1012
23064	101276	006300				ASL	R0	:	
23065	101300	006306				ASL	R0	:	
23066	101302	042700	140377			BIC	#140377,R0	:	
23067	101306	042777	037400	112010		BIC	#37400,@ML.REG+150	:	
23068	101314	050077	112004			BIS	R0,@ML.REG+150	:	
23069	101320	111200				MOVB	(R2),R0	:	1012
23070	101322	042700	177700			BIC	#177700,R0	:	
23071	101326	142777	000077	111770		BICB	#77,@ML.REG+150	:	
23072	101334	150077	111764			BISB	R0,@ML.REG+150	:	
23073	101340	116500	013540			MOVB	DT.1+4(R5),R0	:	1012
23074	101344	042700	177700			BIC	#177700,R0	:	
23075	101350	142777	000077	111756		BICB	#77,@ML.REG+160	:	
23076	101356	150077	111752			BISB	R0,@ML.REG+160	:	
23077	101362	004767	114474			JSR	PC,GD.BLK.XFER	:	1012
23078	101366	142777	000002	111700		BICB	#2,@ML.REG+120	:	1012
23079	101374	012777	000071	111552		MOV	#71,@ML.REG	:	1012
23080	101402	105777	111616		4\$:	TSTB	@ML.REG+50	:	
23081	101406	100375				BPL	4\$:	
23082	101410	011146				MOV	(R1),-(SP)	:	1013
23083	101412	042716	157777			BIC	#157777,(SP)	:	

23085										
23086										
23087										
23088	101416	017700	111742			MOV	@ML.REG+210,R0			
23089	101422	042700	157777			BIC	#157777,R0			
23090	101426	020026				CMP	R0,(SP)+			
23091	101430	001402				BEQ	5\$			
23092	101432	012703	000001			MOV	#1,R3			
23093	101436	011146		5\$:		MOV	(R1),-(SP)		;	* ,ERR.FLG
23094	101440	042716	137777			BIC	#137777,(SP)		;	
23095	101444	017700	111714			MOV	@ML.REG+210,R0			
23096	101450	042700	137777			BIC	#137777,R0			
23097	101454	020026				CMP	R0,(SP)+			
23098	101456	001402				BEQ	6\$			
23099	101460	012703	000001			MOV	#1,R3			
23100	101464	011146		6\$:		MOV	(R1),-(SP)		;	* ,ERR.FLG
23101	101466	042716	077777			BIC	#77777,(SP)		;	
23102	101472	017700	111666			MOV	@ML.REG+210,R0			
23103	101476	042700	077777			BIC	#77777,R0			
23104	101502	020026				CMP	R0,(SP)+			
23105	101504	001402				BEQ	7\$			
23106	101506	012703	000001			MOV	#1,R3			
23107	101512	111146		7\$:		MOVB	(R1),-(SP)		;	* ,ERR.FLG
23108	101514	042716	177700			BIC	#177700,(SP)		;	
23109	101520	117700	111640			MOVB	@ML.REG+210,R0			
23110	101524	042700	177700			BIC	#177700,R0			
23111	101530	020026				CMP	R0,(SP)+			
23112	101532	001402				BEQ	8\$			
23113	101534	012703	000001			MOV	#1,R3			
23114	101540	011146		8\$:		MOV	(R1),-(SP)		;	* ,ERR.FLG
23115	101542	042716	170077			BIC	#170077,(SP)		;	
23116	101546	017700	111612			MOV	@ML.REG+210,R0			
23117	101552	042700	170077			BIC	#170077,R0			
23118	101556	020026				CMP	R0,(SP)+			
23119	101560	001402				BEQ	9\$			
23120	101562	012703	000001			MOV	#1,R3			
23121	101566	011246		9\$:		MOV	(R2),-(SP)		;	* ,ERR.FLG
23122	101570	005046				CLR	-(SP)		;	
23123	101572	032766	010000	000002		BIT	#10000,2(SP)			
23124	101600	001401				BEQ	10\$			
23125	101602	005216				INC	(SP)			
23126	101604	005000				CLR	R0			
23127	101606	132777	000100	111420		BITB	#100,@ML.REG+60			
23128	101614	001401				BEQ	11\$			
23129	101616	005200				INC	R0			
23130	101620	020026				CMP	R0,(SP)+			
23131	101622	001002				BNE	12\$			
23132	101624	005726				TST	(SP)+			
23133	101626	000425				BR	13\$			
23134	101630	005726				TST	(SP)+			
23135	101632	104455				TRAP	55			
23136	101634	000173				.WORD	173		;	1014
23137	101636	010464				.WORD	ASYNC			
23138	101640	024052				.WORD	DUMPER			
23139	101642	012746	006442			MOV	#WORD.14,-(SP)		;	1014

Address	Offset	Hex	Dec	Label	Instruction	Comments	Page
23141				:ML4			
23142				:			
23143				:			
23144	101646	012746	007252		MOV #WRD.69,-(SP)		
23145	101652	012746	006406		MOV #WRD.10,-(SP)		
23146	101656	012746	007240		MOV #WRD.67,-(SP)		
23147	101662	012746	006046		MOV #FOR.FMT,-(SP)		
23148	101666	012746	000005		MOV #5,-(SP)		
23149	101672	010600			MOV SP,R0	: SP,*	
23150	101674	104414			TRAP 14	:	
23151	101676	062706	000014		ADD #14,SP	:	1014
23152	101702	020327	000001	13\$:	CMP R3,#1	: ERR.FLG,*	1014
23153	101706	001062			BNE 14\$:	
23154	101710	104455			TRAP 55	:	1015
23155	101712	000174			.WORD 174	:	
23156	101714	010526			.WORD SYNC	:	
23157	101716	024052			.WORD DUMPER	:	
23158	101720	012746	007576		MOV #PHR.11,-(SP)	:	1015
23159	101724	012746	007066		MOV #WRD.52,-(SP)	:	
23160	101730	012746	010376		MOV #REG.14,-(SP)	:	
23161	101734	012746	006034		MOV #THR.FMT,-(SP)	:	
23162	101740	012746	000004		MOV #4,-(SP)	:	
23163	101744	010600			MOV SP,R0	: SP,*	
23164	101746	104414			TRAP 14	:	
23165	101750	017716	111410		MOV @ML.REG+210,(SP)	:	1015
23166	101754	011146			MOV (R1),-(SP)	:	
23167	101756	012746	004222		MOV #FMT.1,-(SP)	:	
23168	101762	012746	000003		MOV #3,-(SP)	:	
23169	101766	010600			MOV SP,R0	: SP,*	
23170	101770	104414			TRAP 14	:	
23171	101772	011200			MOV (R2),R0	:	1015
23172	101774	006200			ASR R0	:	
23173	101776	006200			ASR R0	:	
23174	102000	006200			ASR R0	:	
23175	102002	006200			ASR R0	:	
23176	102004	006200			ASR R0	:	
23177	102006	006200			ASR R0	:	
23178	102010	042700	177700		BIC #177700,R0	:	
23179	102014	010016			MOV R0,(SP)	:	
23180	102016	111246			MOVB (R2),-(SP)	:	
23181	102020	042716	177700		BIC #177700,(SP)	:	
23182	102024	116546	013540		MOVB DT.1+4(R5),-(SP)	:	
23183	102030	042716	177700		BIC #177700,(SP)	:	
23184	102034	012746	005412		MOV #FMT.19,-(SP)	:	
23185	102040	012746	000004		MOV #4,-(SP)	:	
23186	102044	010600			MOV SP,R0	: SP,*	
23187	102046	104414			TRAP 14	:	
23188	102050	062706	000030		ADD #30,SP	:	1014
23189	102054	104467		14\$:	TRAP 67	:	1015
23190	102056	006000			ROR R0	:	
23191	102060	103002			BHIS 15\$:	
23192	102062	000167	177170		JMP 3\$:	
23193	102066	022626		15\$:	CMP (SP)+,(SP)+	:	1011
23194	102070	005204			INC R4	: LATCH.LOOP	1011
23195	102072	020466	000002	16\$:	CMP R4,2(SP)	: LATCH.LOOP,FINISH	

23197												22-Dec-1980 09:24:31	TOPS
23198												22-Dec-1980 09:21:22	PA:<
23199													
23200	102076	003002					BGT	17\$					
23201	102100	000167	177122				JMP	2\$					
23202	102104	152777	000040	111102	17\$:		BISB	#40,@ML.REG+40	:				1015
23203	102112	016705	111464				MOV	ML,DUT,R5					
23204	102116	042705	177770				BIC	#177770,R5					
23205	102122	142777	000007	111064			BICB	#7,@ML.REG+40					
23206	102130	150577	111060				BISB	R5,@ML.REG+40					
23207	102134	005777	111224				TST	@ML.REG+210	:				1016
23208	102140	001424					BEQ	18\$					
23209	102142	104455					TRAP	5\$:				1016
23210	102144	000175					.WORD	17\$					
23211	102146	010526					.WORD	SYNC					
23212	102150	024052					.WORD	DUMPER					
23213	102152	012746	006442				MOV	#WRD.14,-(SP)	:				1016
23214	102156	012746	010012				MOV	#FNC.7,-(SP)					
23215	102162	012746	007066				MOV	#WRD.52,-(SP)					
23216	102166	012746	010376				MOV	#REG.14,-(SP)					
23217	102172	012746	006046				MOV	#FOR.FMT,-(SP)					
23218	102176	012746	000005				MOV	#5,-(SP)					
23219	102202	010600					MOV	SP,R0	:	SP,*			
23220	102204	104414					TRAP	14					
23221	102206	062706	000014				ADD	#14,SP	:				1016
23222	102212	152777	000001	111054	18\$:		BISB	#1,@ML.REG+120	:				1016
23223	102220	005066	000002				CLR	2(SP)	:	FINISH			1016
23224	102224	005216					INC	(SP)	:	CLR.LOOP			1011
23225	102226	021627	000001				CMP	(SP),#1	:	CLR.LOOP,*			
23226	102232	003002					BGT	19\$					
23227	102234	000167	176760				JMP	1\$					
23228	102240	022626					19\$:	CMP	(SP)+,(SP)+	:			1006
23229	102242	000207					RTS	PC	:				
23230													
23231													
23232													
23237													
23238													
23242													
23246	102244						T56::						
23247	102244	004767	176706				1\$:	JSR	PC,\$T56	:			1017
23248	102250	104466					TRAP	66					
23249	102252	006000					ROR	R0					
23250	102254	103773					3LO	1\$					

: Routine Size: 283 words
 : Maximum stack depth per invocation: 22 words

23252
23253
23254
23255 102256 000207
23256
23257
23258
23263
23264
23265 ; 10173 !<BLF/PAGE>

:ML4
:

RTS PC

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

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

23267 :ML4
23268 :
23269 :
23270 :
23271 :
23272 :
23273 :
23274 :
23275 :
23276 :
23277 :
23278 :
23279 :
23280 :
23281 :
23282 :
23283 :
23284 :
23285 :
23286 :
23287 :
23288 :
23289 :
23290 :
23291 :
23292 :
23293 :
23294 :
23295 :
23296 :
23297 :
23298 :
23299 :
23300 :
23301 :
23302 :
23303 :
23304 :
23305 :
23306 :
23307 :
23308 :
23309 :
23310 :
23311 :
23312 :
23313 :
23314 :
23315 :
23316 :
23317 :
23318 :
23319 :
23320 :
23321 :

10174 !
10175 !
10176 !
10177 !
10178 !
10179 !
10180 !
10181 !
10182 !
10183 !
10184 !
10185 !
10186 !
10187 !
10188 !
10189 !
10190 !
10191 !
10192 !
10193 !
10194 !
10195 !
10196 !
10197 !
10198 !
10199 !
10200 !
10201 !
10202 !
10203 !
10204 !
10205 !
10206 !
10207 !
10208 !
10209 !
10210 !
10211 !
10212 !
10213 !
10214 !
10215 !
10216 !
10217 !
10218 !
10219 !
10220 !
10221 !
10222 !
10223 !
10224 !
10225 !

! BGNTST;

! ++

! TEST NUMBER: TST 57

! TEST NAME: ECC ERROR LOCATION REGISTER TEST

! TEST DESCRIPTION:

! TEST THE ECC ERROR LOCATION REGISTER FOR CLOCKING, BIT UNIQUENESS
! CLEARING AND LATCHING OF DSA ADDRESSES ON DETECTION OF ECC ERROR BY:

1. TEST FOR CLOCKING BY:
 - A. DO A MASS BUS CLEAR FORCING REGISTER BITS TO ZEROES
 - B. FORCE AN ECC ERROR AT THE LAST BLOCK AND DO A READ TRANSFER
 - C. CHECK MLEL FOR LAST BLOCK ADDRESS
2. TEST FOR BIT UNIQUENESS BY:
 - A. WITH FORCED ECC ERRORS DO READ TRANSFERS AT VARIOUS DSA ADDRESSES AND CHECK MLEL FOR LATCHED DSA ADDRESSES
 - B. DO CLEAR AND REPEAT
3. TEST FOR LATCHING AND CLEARING BY:
 - A. WITH A FORCED SINGLE BIT ERROR DO A READ TRANSFER AT THE LAST BLOCK. (LATCHES MLEL WITH THE LAST BLOCK ADDRESS TO START THE TEST OFF)
 - B. AGAIN WITH A FORCED SINGLE BIT ERROR DO A READ TRANSFER AT DSA ZERO AND CHECK MLEL FOR NOT LATCHING.
NO CLEAR DONE
 - C. WITH FORCED UNCORRECTABLE ERRORS DO A READ TRANSFER AT DSA ZERO CHECK MLEL FOR LATCHED DSA ZERO
NO CLEAR DONE
 - D. WITH FORCED SINGLE BIT ERRORS DO A READ TRANSFER AT THE LAST BLOCK AND CHECK MLEL FOR NOT LATCHING
NO CLEAR DONE
 - E. WITH A FORCED UNCORRECTABLE ERROR DO A READ TRANSFER AT THE LAST BLOCK AN CHECK MLEL FOR NOT LATCHING
 - F. DO A MASS BUS CLEAR
 - G. WITH A FORCED SINGLE BIT ERROR DO A READ TRANSFER AT THE LAST BLOCK
 - H. DO A MASS BUS CLEAR AGAIN AND CHECK MLEL FOR ZEROES

! IMPLICIT INPUTS:

! IO_BUF
! A VECTOR OF 256 WORDS WHERE DATA
! FOR MBUS READS AND WRITES TRANSFERS
! IS FOUND.

! local
! SGL,

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (107)

!SINGLE BIT ERROR PARRAMETER

```

23323 :ML4
23324 :
23325 :
23326 :      10226      UNC,
23327 :      10227      SAVE,
23328 :      10228      SHF_DATA,
23329 :      10229      DSA_PAT;
23330 :
23331 :      10231      BGNSUB;
23332 :      10232      CLR_MBUS;
23333 :      10233      FOR_ECC_ERR (SGL = 1, UNC = 0);
23334 :      10234      MLDA = .LST_BLK;
23335 :      10235      MLWC = not 255;
23336 :      10236      MLBA = IO_BUF;
23337 :      10237      MLCS1 = read;
23338 :      10238      TIME_OUT_LOOP;
23339 :
23340 :      10240      if .MLEL eql ZEROES
23341 :      10241      then
23342 :      10242      begin
23343 :      10243      ERRDF (126, SYNC, DUMPER);
23344 :      10244      PRINTB (THR_FMT, REG_15, WRD_52, PHR_11);
23345 :      10245      EXIT_TST;
23346 :      10246      end;
23347 :
23348 :      10248      ENDSUB;
23349 :      10249      SHF_DATA = .LST_ARR;
23350 :
23351 :      10251      incr LOOP from 0 to 16 do
23352 :      10252      begin
23353 :      10253      BGNSUB;
23354 :      10254      CLR_MBUS;
23355 :      10255      FOR_ECC_ERR (SGL = 1, UNC = 0);
23356 :      10256
23357 :      10257      selectone .LOOP of
23358 :      10258      set
23359 :      10259
23360 :      10260      [0] :
23361 :      10261      begin
23362 :      10262      DSA_PAT = ZEROES;
23363 :      10263      MLDA = .DSA_PAT;
23364 :      10264      end;
23365 :      10265
23366 :      10266      [1] :
23367 :      10267      begin
23368 :      10268      DSA_PAT = .LST_BLK;
23369 :      10269      MLDA = .DSA_PAT;
23370 :      10270      end;
23371 :      10271
23372 :      10272      [otherwise] :
23373 :      10273      begin
23374 :      10274      DSA_PAT = .SHF_DATA;
23375 :      10275      MLDA = .SHF_DATA;
23376 :      10276      SHF_DATA = .SHF_DATA^-ONE;
23377 :      10277      end;

```

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:(NEALE)BL3ML4.BLI.2 (107)

```

!UNCORRECTABLE ERR PARRAMETER
!TEMPORY SAVE LOCATION
!SHIFTING DATA PATTERN SAVE LOCATION
!EXPECTED DSA PATTERN LATCHED IN MLEL

!START OF SCOPE LOOP
!CLEAR THE MLEL REGISTER
!FORCE A SBE
!LATCH THE LAST BLK ADRS INTO THE MLEL

!DO A READ AND LATCH THE MLEL

!SEE IF THE MLEL GOT CLOCKED

!REPORT AN ERROR IF NOT CLOCKED

!END OF SCOPE LOOP
!LOAD THE SHIFT DATA

!FIRST TEST FOR 1'S AND 0'S THEN SHIFT DATA

!START OF SCOPE LOOP
!CLEAR THE DRIVE
!FORCE A SBE

!SELECT THE TEST PATTERN

!ZEROES PATTERN

!MOSTLY ONES PATTERN

!SHIFT PATTERN

```

```

23379 :ML4
23380 :
23381 :
23382 : 10278 tes;
23383 : 10279
23384 : 10280 MLWC = not 255; !SET UP THE TRANSFER
23385 : 10281 MLBA = IO_BUF;
23386 : 10282 MLCS1 = read; !DO THE TRANSFER AND LATCH MLEL
23387 : 10283 TIME_OUT_LOOP;
23388 : 10284 SAVE = .MLEL; !READ AND SAVE THE MLEL REGISTER
23389 : 10285
23390 : 10286 if .SAVE neq .DSA_PAT !SEE IF SAVE IS WHAT WE EXPECTED
23391 : 10287 then
23392 : 10288 begin !REPORT AN ERROR IF NOT THE SAME
23393 : 10289 ERRDF (127, ASYNC, DUMPER);
23394 : 10290 PRINTB (TWO_FMT, REG 15, PHR 4);
23395 : 10291 PRINTB (FMT_2, .DSA_PAT, .SAVE, (.DSA_PAT xor .SAVE));
23396 : 10292 end;
23397 : 10293
23398 : 10294 ENDSUB; !END OF SCOPE LOOP
23399 : 10295 end;
23400 : 10296
23401 : 10297 CLR_MBUS; !CLEAR THE DRIVE
23402 : 10298 FOR_ECC_ERR (SGL = 1, UNC = 0); !FORCE A SBE
23403 : 10299 MLWC = not 255; !SET UP A LAST BLOCK TRANSFER
23404 : 10300 MLBA = IO_BUF; !TO LATCH THE MLEL WITH T
23405 : 10301 MLDA = .LST_BLK;
23406 : 10302 MLCS1 = read;
23407 : 10303 TIME_OUT_LOOP;
23408 : 10304
23409 : 10305 incr LOOP from 0 to 4 do !NOW TEST FOR CORRECT LATCHING
23410 : 10306 begin
23411 : 10307 BGNSUB; !START OF SCOPE LOOP
23412 : 10308 MLER = ZEROES; !CLEAR THE ERROR REGISTER
23413 : 10309
23414 : 10310 case .LOOP from 0 to 4 of !SELECT TEST DATA
23415 : 10311 set
23416 : 10312
23417 : 10313 [0] : !FORCE A SBE AND CHECK FOR THE LST BLK ADRS
23418 : 10314 begin
23419 : 10315 SGL = 1;
23420 : 10316 UNC = 0;
23421 : 10317 DSA_PAT = .LST_BLK;
23422 : 10318 MLDA = ZEROES;
23423 : 10319 end;
23424 : 10320
23425 : 10321 [1] : !FORCE A UNC ERROR AND CHECK FOR ADRS ZERO
23426 : 10322 begin
23427 : 10323 SGL = 0;
23428 : 10324 UNC = 1;
23429 : 10325 DSA_PAT = ZEROES;
23430 : 10326 MLDA = ZEROES;
23431 : 10327 end;
23432 : 10328
23433 : 10329 [2] : !FORCE SBE AND CHECK FOR ADRS ZERO

```

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA: <NEALE>BL3ML4.BLI.2 (107)

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (107)

```

23435 :ML4
23436 :
23437 :
23438 :      10330      begin
23439 :      10331      SGL = 1;
23440 :      10332      UNC = 0;
23441 :      10333      DSA PAT = ZEROES;
23442 :      10334      MLDA = .LST_BLK;
23443 :      10335      end;
23444 :      10336
23445 :      10337      [3] :      !FORCE UNC ERROR AND CHECK FOR ADRS ZERO
23446 :      10338      begin
23447 :      10339      SGL = 0;
23448 :      10340      UNC = 1;
23449 :      10341      DSA PAT = ZEROES;
23450 :      10342      MLDA = .LST_BLK;
23451 :      10343      end;
23452 :      10344
23453 :      10345      [4] :      !FORCE SBE AND CHECK FOR ADRS ZERO
23454 :      10346      begin
23455 :      10347      CLR_MBUS;
23456 :      10348      SGL = 1;
23457 :      10349      UNC = 0;
23458 :      10350      DSA PAT = ZEROES;
23459 :      10351      MLDA = .LST_BLK;
23460 :      10352      end;
23461 :      10353      tes;
23462 :      10354
23463 :      10355      FOR ECC_ERR (.SGL, .UNC);
23464 :      10356      MLWC = not 255;
23465 :      10357      MLBA = IO_BUF;
23466 :      10358      MLCS1 = read;
23467 :      10359      TIME_OUT_LOOP;
23468 :      10360
23469 :      10361      if .LOOP neq 4
23470 :      10362      then
23471 :      10363          SAVE = .MLEL
23472 :      10364      else
23473 :      10365          begin
23474 :      10366          CLR_MBUS;
23475 :      10367          SAVE = .MLEL;
23476 :      10368          end;
23477 :      10369
23478 :      10370      if .SAVE neq .DSA_PAT
23479 :      10371      then
23480 :      10372          begin
23481 :      10373          ERRDF (128, SYNC, DUMPER);
23482 :      10374          PRINTB (THR_FMT, REG 15, WRD 71, WRD_10);
23483 :      10375          PRINTB (FMT_1, .DSA_PAT, .SAVE)
23484 :      10376          end;
23485 :      10377
23486 :      10378      ENDSUB;
23487 :      10379      end;
23488 :      10380
23489 :      10381      ENDTST;

```

!FORCE SELECTED FORCED ERROR INTO ML11
!SET UP A TRANSFER
!READ THE ERROR OUT AND LATCH MLEL
!TEST THE REG FOR LATCH DATA IF LOOP NEQ 4
!ELSE TEST THE REG FOR CLEAR
!SEE IF THE REG LATCHED OK
!REPORT ERROR IF NOT LATCHED
!END OF SCOPE LOOP

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA: <NEALE>BL3ML4.BLI.2 (107)

Address	Offset	Hex	Hex	Label	Instruction	Comments	Address
23491							
23492							
23493							
23497							
23501	102260	004167	101632	\$T57:	JSR R1,\$\$SAVE5	:	1017
23502	102264	005746			TST -(SP)	:	
23503	102266	104402		1\$:	TRAP 2	:	1022
23504	102270	152777	000040	110716	BISB #40,@ML.REG+40	:	1023
23505	102276	070705	111300		MOV ML.DUT,R5	:	
23506	102302	042705	177770		BIC #177770,R5	:	
23507	102306	142777	000007	110700	BICB #7,@ML.REG+40	:	
23508	102314	150577	110674		BISB R5,@ML.REG+40	:	
23509	102320	012703	000001		MOV #1,R3	: *,SGL	1023
23510	102324	010346			MOV R3,-(SP)	: SGL,*	
23511	102326	005001			CLR R1	: UNC	
23512	102330	005046			CLR -(SP)	:	
23513	102332	004767	113422		JSR PC,FOR.ECC.ERR	:	
23514	102336	016777	107230	110640	MOV LST.BLK,@ML.REG+30	:	1023
23515	102344	012777	177400	110612	MOV #-400,@ML.REG+10	:	1023
23516	102352	012777	011600	110614	MOV #10.BUF,@ML.REG+20	:	1023
23517	102360	012777	000071	110566	MOV #71,@ML.REG	:	1023
23518	102366	105777	110632	2\$:	TSTB @ML.REG+50	:	
23519	102372	100775			BPL 2\$:	
23520	102374	005777	110774		TST @ML.REG+220	:	1024
23521	102400	001025			BNE 3\$:	
23522	102402	104455			TRAP 55	:	1024
23523	102404	000176			.WORD 176	:	
23524	102406	010526			.WORD SYNC	:	
23525	102410	024052			.WORD DUMPER	:	
23526	102412	012746	007576		MOV #PHR.11,-(SP)	:	1024
23527	102416	012746	007066		MOV #WRD.52,-(SP)	:	
23528	102422	012746	010404		MOV #REG.15,-(SP)	:	
23529	102426	012746	006034		MOV #THR.FMT,-(SP)	:	
23530	102432	012746	000004		MOV #4,-(SP)	:	
23531	102436	010600			MOV SP,R0	: SP,*	
23532	102440	104414			TRAP 14	:	
23533	102442	104463			TRAP 63	:	
23534	102444	062706	000016		ADD #16,SP	:	1024
23535	102450	000167	001006		JMP 26\$:	1024
23536	102454	022626		3\$:	CMP (SP)+,(SP)+	:	1022
23537	102456	104467			TRAP 67	:	1024
23538	102460	006000			ROR R0	:	
23539	102462	103701			BLO 1\$:	
23540	102464	016705	107106		MOV LST.ARR,R5	: *,SHF.DATA	1024
23541	102470	005002			CLR R2	: LOOP	1025
23542	102472	104402		4\$:	TRAP 2	:	1025
23543	102474	152777	000040	110512	BISB #40,@ML.REG+40	:	1025
23544	102502	016700	111074		MOV ML.DUT,R0	:	

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Line No.
23546				:ML4			
23547				:			
23548							
23549	102506	042700	177770		BIC #177770,R0		
23550	102512	142777	000007	110474	BICB #7,@ML.REG+40		
23551	102520	150077	110470		BISB R0,@ML.REG+40		
23552	102524	012703	000001		MOV #1,R3	: *,SGL	1025
23553	102530	010346			MOV R3,-(SP)	: SGL,*	
23554	102532	005001			CLR R1	: UNC	
23555	102534	005046			CLR -(SP)		
23556	102536	004767	113216		JSR PC,FOR.ECC.ERR		
23557	102542	005702			TST R2	: LOOP	1025
23558	102544	001002			BNE 5\$		
23559	102546	005004			CLR R4	: DSA.PAT	1026
23560	102550	000405			BR 6\$		1026
23561	102552	020227	000001	5\$:	CMP R2,#1	: LOOP,*	1025
23562	102556	001005			BNE 7\$		
23563	102560	016704	107006		MOV LST.BLK,R4	: *,DSA.PAT	1026
23564	102564	010477	110414	6\$:	MOV R4,@ML.REG+30	: DSA.PAT,*	1026
23565	102570	000404			BR 8\$		1025
23566	102572	010504		7\$:	MOV R5,R4	: SHF.DATA,DSA.PAT	1027
23567	102574	010577	110404		MOV R5,@ML.REG+30	: SHF.DATA,*	1027
23568	102600	006205			ASR R5	: SHF.DATA	1027
23569	102602	012777	177400	110354	MOV #-400,@ML.REG+10		1028
23570	102610	012777	011600	110356	MOV #10.BUF,@ML.REG+20		1028
23571	102616	012777	000071	110330	MOV #71,@ML.REG		1028
23572	102624	105777	110374	9\$:	TSTB @ML.REG+50		
23573	102630	100375			BPL 9\$		
23574	102632	017766	110536	000004	MOV @ML.REG+220,4(SP)	: *,SAVE	1028
23575	102640	026604	000004		CMP 4(SP),R4	: SAVE,DSA.PAT	1028
23576	102644	001441			BEQ 10\$		
23577	102646	104455			TRAP 55		1028
23578	102650	000177			.WORD 177		
23579	102652	010464			.WORD ASYNC		
23580	102654	024052			.WORD DUMPER		
23581	102656	012746	007454		MOV #PHR.4,-(SP)		1029
23582	102662	012746	010404		MOV #REG.15,-(SP)		
23583	102666	012746	006024		MOV #TWO.FMT,-(SP)		
23584	102672	012746	000003		MOV #3,-(SP)		
23585	102676	010600			MOV SP,R0	: SP,*	
23586	102700	104414			TRAP 14		
23587	102702	016616	000014		MOV 14(SP),(SP)	: SAVE,*	1029
23588	102706	010446			MOV R4,-(SP)	: DSA.PAT,*	
23589	102710	046616	000002		BIC 2(SP),(SP)		
23590	102714	040466	000002		BIC R4,2(SP)	: DSA.PAT,*	
23591	102720	052616			BIS (SP)+(SP)		
23592	102722	016646	000014		MOV 14(SP),-(SP)	: SAVE,*	
23593	102726	010446			MOV R4,-(SP)	: DSA.PAT,*	
23594	102730	012746	004266		MOV #FMT.2,-(SP)		
23595	102734	012746	000004		MOV #4,-(SP)		
23596	102740	010600			MOV SP,R0	: SP,*	
23597	102742	104414			TRAP 14		
23598	102744	062706	000020		ADD #20,SP		1028
23599	102750	022626		10\$:	CMP (SP)+(SP)+		1025
23600	102752	104467			TRAP 67		1029

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

Address	Hex	Hex	Hex	Label	Instruction	Comments	Line
23602							
23603							
23604							
23605	102754	006000			ROR R0		
23606	102756	103645			BLO 4\$		
23607	102760	005202			INC R2	: LOOP	1025
23608	102762	020227	000020		CMP R2,#20	: LOOP,*	
23609	102766	003641			BLE 4\$		
23610	102770	152777	000040	110216	BISB #40,@ML.REG+40	:	1029
23611	102776	016705	110600		MOV ML.DUT,R5		
23612	103002	042705	177770		BIC #177770,R5		
23613	103006	142777	000007	110200	BICB #7,@ML.REG+40		
23614	103014	150577	110174		BISB R5,@ML.REG+40		
23615	103020	012703	000001		MOV #1,R3	: *,SGL	1029
23616	103024	010346			MOV R3,-(SP)	: SGL,*	
23617	103026	005001			CLR R1	: UNC	
23618	10 30	005046			CLR -(SP)		
23619	10 32	004767	112722		JSR PC,FOR.ECC.ERR		
23620	10 36	012777	177400	110120	MOV #400,@ML.REG+10	:	1029
23621	10 44	012777	011600	110122	MOV #10.BUF,@ML.REG+20	:	1030
23622	10 52	016777	106514	110124	MOV LST.BLK,@ML.REG+30	:	1030
23623	10 56	012777	000071	110066	MOV #71,@ML.REG	:	1030
23624	103066	105777	110132		MOV @ML.REG+50		
23625	103072	100375		11\$:	TSTB 11\$		
23626	103074	005002			CLR R2	: LOOP	1030
23627	103076	104402		12\$:	TRAP 2	:	1030
23628	103100	005077	110130		CLR @ML.REG+60	:	1030
23629	103104	010205			MOV R2,R5	: LOOP,*	1031
23630	103106	006305			ASL PC		
23631	103110	066507	103114		ADD 3\$(R5),PC		
23632	103114	000012		13\$:	.WORD 14\$-13\$		
23633	103116	000026			.WORD 15\$-13\$		
23634	103120	000044			.WORD 17\$-13\$		
23635	103122	000046			.WORD 18\$-13\$		
23636	103124	000056			.WORD 19\$-13\$		
23637	103126	012703	000001	14\$:	MOV #1,R3	: *,SGL	1031
23638	103132	005001			CLR R1	: UNC	1031
23639	103134	016704	106432		MOV LST.BLK,R4	: *,DSA.PAT	1031
23640	103140	000404			BR 16\$:	1031
23641	103142	005003		15\$:	CLR R3	: SGL	1032
23642	103144	012701	000001		MOV #1,R1	: *,UNC	1032
23643	103150	005004			CLR R4	: DSA.PAT	1032
23644	103152	005077	110026	16\$:	CLR @ML.REG+30	:	1032
23645	103156	000430			BR 22\$:	1031
23646	103160	000420		17\$:	BR 20\$:	1033
23647	103162	005003		18\$:	CLR R3	: SGL	1033
23648	103164	012701	000001		MOV #1,R1	: *,UNC	1034
23649	103170	000417			BR 21\$:	1034
23650	103172	152777	000040	110014	BISB #40,@ML.REG+40	:	1034
23651	103200	016705	110376		MOV ML.DUT,R5		
23652	103204	042705	177770		BIC #177770,R5		
23653	103210	142777	000007	107776	BICB #7,@ML.REG+40		
23654	103216	150577	107772		BISB R5,@ML.REG+40		
23655	103222	012703	000001	20\$:	MOV #1,R3	: *,SGL	1034
23656	103226	005001			CLR R1	: UNC	1034

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:4

23658									22-Dec-1980 09:24:31	TOPS
23659									22-Dec-1980 09:21:22	PA:4
23660										
23661	103230	005004			21\$:	CLR	R4	:	DSA.PAT	1035
23662	103232	016777	106334	107744		MOV	LST.BLK,@ML.REG+30	:		1035
23663	103240	010346			22\$:	MOV	R3,-(SP)	:	SGL,*	1035
23664	103242	010146				MOV	R1,-(SP)	:	UNC,*	
23665	103244	004767	112510			JSR	PC,FOR.ECC.ERR	:		
23666	103250	012777	177400	107706		MOV	#400,@ML.REG+10	:		1035
23667	103256	012777	011600	107710		MOV	#10.BUF,@ML.REG+20	:		1035
23668	103264	012777	000071	107662		MOV	#71,@ML.REG	:		1035
23669	103272	105777	107726		23\$:	TSTB	@ML.REG+50	:		
23670	103276	100375				BPL	23\$:		
23671	103300	020227	000004			CMP	R2,#4	:	LOOP,*	1036
23672	103304	001014				BNE	24\$:		1036
23673	103306	152777	000040	107700		BISB	#40,@ML.REG+40	:		1036
23674	103314	016705	110262			MOV	ML,DUT,R5	:		
23675	103320	042705	177770			BIC	#177770,R5	:		
23676	103324	142777	000007	107662		BICB	#7,@ML.REG+40	:		
23677	103332	150577	107656			BISB	R5,@ML.REG+40	:		
23678	103336	017766	110032	000010	24\$:	MOV	@ML.REG+220,10(SP)	:	*,SAVE	1036
23679	103344	026604	000010			CMP	10(SP),R4	:	SAVE,DSA.PAT	1037
23680	103350	001433				BEQ	25\$:		
23681	103352	104455				TRAP	55	:		1037
23682	103354	000200				.WORD	200	:		
23683	103356	010526				.WORD	SYNC	:		
23684	103360	024052				.WORD	DUMPER	:		
23685	103362	012746	006406			MOV	#WORD.10,-(SP)	:		1037
23686	103366	012746	007272			MOV	#WORD.71,-(SP)	:		
23687	103372	012746	010404			MOV	#REG.15,-(SP)	:		
23688	103376	012746	006034			MOV	#THR.FMT,-(SP)	:		
23689	103402	012746	000004			MOV	#4,-(SP)	:		
23690	103406	010600				MOV	SP,R0	:	SP,*	
23691	103410	104414				TRAP	14	:		
23692	103412	016616	000022			MOV	22(SP),(SP)	:	SAVE,*	1037
23693	103416	010446				MOV	R4,-(SP)	:	DSA.PAT,*	
23694	103420	012746	004222			MOV	#FMT.1,-(SP)	:		
23695	103424	012746	000003			MOV	#3,-(SP)	:		
23696	103430	010600				MOV	SP,R0	:	SP,*	
23697	103432	104414				TRAP	14	:		
23698	103434	062706	000020			ADD	#20,SP	:		1037
23699	103440	022626			25\$:	CMP	(SP)+,(SP)+	:		1030
23700	103442	104467				TRAP	67	:		1037
23701	103444	006000				ROR	R0	:		
23702	103446	103613				BLO	12\$:		
23703	103450	005202				INC	R2	:	LOOP	1030
23704	103452	020227	000004			CMP	R2,#4	:	LOOP,*	
23705	103456	003607				BLE	12\$:		
23706	103460	022626				CMP	(SP)+,(SP)+	:		
23707	103462	005726			26\$:	TST	(SP)+	:		1017
23708	103464	000207				RTS	PC	:		
23709										
23710										
23711										

; Routine Size: 323 words
; Maximum stack depth per invocation: 19 words

23720
23721
23725
23729 103466
23730 103466 004767 176566
23731 103472 104466
23732 103474 006000
23733 103476 103773
23734 103500 000207
23735
23736
23737
23742
23743
23744 ;

T57::
1\$: JSR PC,\$T57 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

1037

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

10382 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (108)

```

23746 :ML4
23747 :
23748 :
23749 : 10383 !
23750 : 10384 BGNTST;
23751 : 10385
23752 : 10386 !++
23753 : 10387 TEST NUMBER: TST 58
23754 : 10388
23755 : 10389 TEST NAME: ECC HARD ERROR BIT TEST
23756 : 10390
23757 : 10391 TEST DESCRIPTION:
23758 : 10392 VIA ECC_DM AND ECC_DIS TEST THE
23759 : 10393 ECH BIT FOR SETTING AND NOT SETTING BY:
23760 : 10394
23761 : 10395 1. VIA ECC_DM AND A WRITE TRANSFER FORCE
23762 : 10396 ECC ERRORS INTO THE ML11'S GOOD BLOCK
23763 : 10397
23764 : 10398 2. WITH ECC_DM = 0 AND ECC_DIS = 0
23765 : 10399 DO A READ TRANSFER AND CHECK ECH SET
23766 : 10400
23767 : 10401 3. WITH ECC_DM = 0 AND ECC_DIS = 1
23768 : 10402 DO A READ TRANSFER AND CHECK ECH NOT SET
23769 : 10403
23770 : 10404 IMPLICIT INPUTS:
23771 : 10405 IO_BUF
23772 : 10406 A VECTOR OF 256 WORDS WHERE DATA
23773 : 10407 FOR MBUS READS AND WRITES TRANSFERS
23774 : 10408 IS FOUND.
23775 : 10409
23776 : 10410
23777 : 10411 --
23778 : 10412
23779 : 10413 Local
23780 : 10414 EXPECTED; !EXPECTED ECH_ERR VALUE
23781 : 10415
23782 : 10416 CLR_MBUS; !CLEAR THE DRIVE
23783 : 10417 ECC_DM = ONE; !SET ECC DIAG MODE
23784 : 10418 PAR_CRC_WRD = ZEROES; !FORCE UNCORRECTABLE ERRORS INTO THE GOOD BLOCK
23785 : 10419 CRC_A = ZEROES;
23786 : 10420 CRC_B = ZEROES;
23787 : 10421 BAI = ONE;
23788 : 10422 IO_BUF = ONES;
23789 : 10423 GD_BLK_XFER ();
23790 : 10424 MLCS1 = write;
23791 : 10425 TIME_OUT_LOOP;
23792 : 10426
23793 : 10427 incr LOOP from 0 to 1 do !TEST ECH BIT FOR SETTING / NOT SETTING
23794 : 10428 begin
23795 : 10429 BGNSUB; !START OF SCOPE LOOP
23796 : 10430 CLR_MBUS; !CLEAR THE DRIVE
23797 : 10431 GD_BLK_XFER (); !SET UP A GOOD BLOCK TRANSFER
23798 : 10432
23799 : 10433 if .LOOP eql ZERO
23800 : 10434 then !SELECT THE TEST DATA

```

```
23802 :ML4
23803 :
23804 :
23805 :      10435      begin
23806 :      10436      ECC_DIS = 0;
23807 :      10437      EXPECTED = 1;
23808 :      10438      end
23809 :      10439      else
23810 :      10440      begin
23811 :      10441      ECC_DIS = 1;
23812 :      10442      EXPECTED = 0;
23813 :      10443      end;
23814 :      10444
23815 :      10445      MLCS1 = read;
23816 :      10446      TIME_OUT_LOOP;
23817 :      10447
23818 :      10448      if .ECH_ERR neq .EXPECTED
23819 :      10449      then
23820 :      10450      begin
23821 :      10451      ERRDF (129, SYNC, DUMPER);
23822 :      10452      PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_69, WRD_14);
23823 :      10453      PRINTB (FMT_1, .EXPECTED, .ECH_ERR);
23824 :      10454      end;
23825 :      10455
23826 :      10456      ENDSUB;
23827 :      10457      end;
23828 :      10458
23829 :      10459      ENDTST;
23830 :
23831 :
23832 :
23833 :
23834 :
23835 :
23836 :
23837 103502 004167 100352      $T58: JSR R1,SSAVE3
23838 103506 152777 000040 107500 BISR A:0,AHL,REG+40
23839 103514 016702 110062      MOV ML,DUT,R2
23840 103520 042702 177770      BIC #177770,R2
23841 103524 142777 000007 107462 BICB #7,AHL,REG+40
23842 103532 150277 107456 BISR R2,AHL,REG+40
23843 103536 152777 000001 107530 BISR #1,AHL,REG+120
23844 103544 042777 037400 107552 BIC #37400,AHL,REG+150
23845 103552 142777 000077 107544 BICB #77,AHL,REG+150
23846 103560 142777 000077 107546 BICB #77,AHL,REG+160
23847 103566 152777 000010 107420 BISR #10,AHL,REG+40
23848 103574 012767 177777 105776 MOV A-1,IO.BUF
23849 103602 004767 112254 JSR PC,GD,BLK.XFER
23850 103606 012777 000061 107340 MOV #61,AHL,REG
23851 103614 105777 107404 1$: TSTB AHL,REG+50
23852 103620 100375      BPL 1$
23853 103622 005001      CLR R1 ; LOOP
23854 103624 104402      TRAP 2 ;
23855 103626 152777 000040 107360 BISR #40,AHL,REG+40 ;
```

23857					:ML4							
23858					:							
23859					:							
23860	103634	016700	107742			MOV	ML,DUT,RO					
23861	103640	042700	177770			BIC	#177770,RO					
23862	103644	142777	000007	107342		BICB	#7,@ML.REG+40					
23863	103652	150077	107336			BISB	RO,@ML.REG+40					
23864	103656	004767	112200			JSR	PC,GD.BLK.XFER					
23865	103662	005701				TST	R1		:	LOOP		1043
23866	103664	001006				BNE	3\$:			1043
23867	103666	142777	000002	107400		BICB	#2,@ML.REG+120		:			1043
23868	103674	012702	000001			MOV	#1,R2		:	*,EXPECTED		1043
23869	103700	000404				BR	4\$:			1043
23870	103702	152777	000002	107364	3\$:	BISB	#2,@ML.REG+120		:			1044
23871	103710	005002				CLR	R2		:	EXPECTED		1044
23872	103712	012777	000071	107234	4\$:	MOV	#71,@ML.REG		:			1044
23873	103720	105777	107300		5\$:	TSTB	@ML.REG+50		:			
23874	103724	100375				BPL	5\$:			
23875	103726	010203				MOV	R2,R3		:	EXPECTED,*		1044
23876	103730	005000				CLR	RO		:			
23877	103732	132777	000100	107274		BITB	#100,@ML.REG+60		:			
23878	103740	001401				BEQ	6\$:			
23879	103742	005200				INC	RO		:			
23880	103744	020003			6\$:	CMP	RO,R3		:			
23881	103746	001441				BEQ	8\$:			
23882	103750	104455				TRAP	55		:			1045
23883	103752	000201				.WORD	201		:			
23884	103754	010526				.WORD	SYNC		:			
23885	103756	024052				.WORD	DUMPER		:			
23886	103760	012746	006442			MOV	#WRD.14,-(SP)		:			1045
23887	103764	012746	007252			MOV	#WRD.69,-(SP)		:			
23888	103770	012746	006406			MOV	#WRD.10,-(SP)		:			
23889	103774	012746	007240			MOV	#WRD.67,-(SP)		:			
23890	104000	012746	006046			MOV	#FOR.FMT,-(SP)		:			
23891	104004	012746	000005			MOV	#5,-(SP)		:			
23892	104010	010600				MOV	SP,RO		:	SP,*		
23893	104012	104414				TRAP	14		:			
23894	104014	005016				CLR	(SP)		:			1045
23895	104016	132777	000100	107210		BITB	#100,@ML.REG+60		:			
23896	104024	001401				BEQ	7\$:			
23897	104026	005216				INC	(SP)		:			
23898	104030	010246			7\$:	MOV	R2,-(SP)		:	EXPECTED,*		
23899	104032	012746	004222			MOV	#FMT.1,-(SP)		:			
23900	104036	012746	000003			MOV	#3,-(SP)		:			
23901	104042	010600				MOV	SP,RO		:	SP,*		
23902	104044	104414				TRAP	14		:			
23903	104046	062706	000022			ADD	#22,SP		:			1045
23904	104052	104467			8\$:	TRAP	67		:			1045
23905	104054	006000				ROR	RO		:			
23906	104056	103662				BLO	2\$:			
23907	104060	005201				INC	R1		:	LOOP		1042
23908	104062	020127	000001			CMP	R1,#1		:	LOOP,*		
23909	104066	003656				BLE	2\$:			
23910	104070	000207				RTS	PC		:			1038
23911									:			

```
23913      ;ML4
23914      ;
23915      ;
23916      ; Routine Size: 124 words
23917      ; Maximum stack depth per invocation: 13 words
23922
23923
23927
23931 104072 T58::
23932 104072 004767 177404 1$: JSR PC,$T58 ;
23933 104076 104466 TRAP 66
23934 104100 006000 ROR R0
23935 104102 103773 BLO 1$
23936 104104 000207 RTS PC
23937
23938      ; Routine Size: 6 words
23939      ; Maximum stack depth per invocation: 0 words
23944
23945
23946 :      10460 !<BLF/PAGE>
```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

1045

23948 :ML4

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (109)

```

23951 : 10461 !
23952 : 10462 BGNTST;
23953 : 10463
23954 : 10464 !++
23955 : 10465 TEST NUMBER: TST 59
23956 : 10466
23957 : 10467 TEST NAME: DATA CHECK ERROR BIT TEST
23958 : 10468
23959 : 10469 TEST DESCRIPTION:
23960 : 10470 VIA ECC_EN, ECC_DM AND ECC_DM TEST THE
23961 : 10471 DCK BIT FOR SETTING AND NOT SETTING BY:
23962 : 10472
23963 : 10473 1. VIA ECC_DM AND A WRITE TRANSFER FORCE
23964 : 10474 ECC ERRORS INTO THE ML11'S GOOD BLOCK
23965 : 10475
23966 : 10476 2. WITH DCK_EN = 1, ECC_DM = 0, ECC_DIS = 0
23967 : 10477 DO A READ TRANSFER AND CHECK THE DCK BIT SET
23968 : 10478
23969 : 10479 3. WITH DCK_EN = 0, ECC_DM = 0, ECC_DIS = 0
23970 : 10480 DO A READ TRANSFER AND CHECK THE DCK BIT NOT SET
23971 : 10481
23972 : 10482 4. WITH ECK_EN = 1, ECC_DM = 0, ECC_DIS = 1
23973 : 10483 DO A READ TRANSFER AND CHECK THE DCK BIT SET
23974 : 10484
23975 : 10485
23976 : 10486 IMPLICIT INPUTS:
23977 : 10487 IO_BUF
23978 : 10488 A VECTOR OF 256 WORDS WHERE DATA
23979 : 10489 FOR MBUS READS AND WRITES TRANSFERS
23980 : 10490 IS FOUND.
23981 : 10491
23982 : 10492
23983 : 10493 --
23984 : 10494
23985 : 10495 local
23986 : 10496 SGL,
23987 : 10497 UNC,
23988 : 10498 EXPECTED;
23989 : 10499
23990 : 10500 CLR_MBUS;
23991 : 10501 FOR_ECC_ERR (SGL = 1, UNC = 0);
23992 : 10502 BAI = ONE;
23993 : 10503 IO_BUF = ZEROES;
23994 : 10504 GD_BLK_XFER ();
23995 : 10505 MLCS1 = write;
23996 : 10506 TIME_OUT_LOOP;
23997 : 10507
23998 : 10508 incr LOOP from 0 to 2 do
23999 : 10509 begin
24000 : 10510 BGNSUB;
24001 : 10511 CLR_MBUS;
24002 : 10512 GD_BLK_XFER ();

```

!SINGLE BIT ERROR PARRAMETER
!UNCORRECTABLE ERROR PARRAMETER
!EXPECTED DATA CHECK BIT VALUE

!CLEAR THE DRIVE
!FORCE A SBE

!SET UP A GOOD BLOCK TRANSFER
!WRITE THE DRIVE WITH THE ERROR

!TEST DCK BIT FOR 3 CONDITIONS

!START OF SCOPE LOOP
!CLEAR THE DRIVE
!SET UP A GOOD BLOCK TRANSFER

24004	:	ML4		22-Dec-1980 09:24:31	TOPS-20 Bliss-16 V2(212)
24005	:			22-Dec-1980 09:21:22	PA:<NEALE>BL3ML4.BLI.2 (109)
24006	:				
24007	:	10513			
24008	:	10514	case .LOOP from 0 to 2 of	!	SELECT THE TEST DATA
24009	:	10515	set		
24010	:	10516			
24011	:	10517	[0] :	!	EXPECT THE DCK BIT TO SET
24012	:	10518	begin		
24013	:	10519	DCK_EN = 1;		
24014	:	10520	ECC_DIS = 0;		
24015	:	10521	EXPECTED = 1;		
24016	:	10522	end;		
24017	:	10523			
24018	:	10524	[1] :	!	EXPECT THE DCK BIT NOT TO SET
24019	:	10525	begin		
24020	:	10526	DCK_EN = 0;		
24021	:	10527	ECC_DIS = 0;		
24022	:	10528	EXPECTED = 0;		
24023	:	10529	end;		
24024	:	10530			
24025	:	10531	[2] :	!	EXPECT THE DCK BIT TO SET
24026	:	10532	begin		
24027	:	10533	DCK_EN = 1;		
24028	:	10534	ECC_DIS = 1;		
24029	:	10535	EXPECTED = 1;		
24030	:	10536	end;		
24031	:	10537	tes;		
24032	:	10538			
24033	:	10539	MLCS1 = read;	!	READ THE ERROR AND CLOCK THE DCK BIT
24034	:	10540	TIME_OUT_LOOP;		
24035	:	10541			
24036	:	10542	if .DCK neq .EXPECTED	!	IS THE DCK BIT WHAT WE EXPECTED
24037	:	10543	then		
24038	:	10544	begin	!	REPORT THE ERRRO IF NOT THE SAME
24039	:	10545			
24040	:	10546	if .LOOP neq ONE then ERRDF (130, SYNC, DUMPER) else ERRDF (130, ASYNC, DUMPER);		
24041	:	10547			
24042	:	10548		!	CHOOSE WHICH THE ERROR IS ON
24043	:	10549	PRINTB (THR_FMT, WRD 72, PHR 5, WRD_10);		
24044	:	10550	PRINTB (FMT_1, .EXPECTED, .DCK);		
24045	:	10551	end;		
24046	:	10552			
24047	:	10553	ENDSUB;	!	END OF THE SCOPE LOOP
24048	:	10554	end;		
24049	:	10555			
24050	:	10556	ENDTST;		
24054	:				

Address	Hex	Hex	Hex	Label	Code	Comment	Line
24059					:ML4		
24060					:		
24061							
24062	104106	004167	077746	107074	\$T59:	JSR R1,SSAVE3	1045
24063	104112	152777	000040			BISB #40,@ML.REG+40	1049
24064	104120	016702	107456			MOV ML,DUT,R2	
24065	104124	042702	177770			BIC #177770,R2	
24066	104130	142777	000007	107056		BICB #7,@ML.REG+40	
24067	104136	150277	107052			BISB R2,@ML.REG+40	
24068	104142	012701	000001			MOV #1,R1	
24069	104146	010146				MOV R1,-(SP)	:*,SGL
24070	104150	005001				CLR R1	:SGL,*
24071	104152	005046				CLR -(SP)	:UNC
24072	104154	004767	111600			JSR PC,FOR.ECC.ERR	
24073	104160	152777	000010	107026		BISB #10,@ML.REG+40	1050
24074	104166	005067	105406			CLR IO.BUF	1050
24075	104172	004767	111664			JSR PC,GD.BLK.XFER	1050
24076	104176	012777	000061	106750		MOV #61,@ML.REG	1050
24077	104204	105777	107014		1\$:	TSTB @ML.REG+50	
24078	104210	100375				BPL 1\$	
24079	104212	005001				CLR R1	: LOOP
24080	104214	104402			2\$:	TRAP 2	
24081	104216	152777	000040	106770		BISB #40,@ML.REG+40	1050
24082	104224	016700	107352			MOV ML,DUT,R0	1051
24083	104230	042700	177770			BIC #177770,R0	
24084	104234	142777	000007	106752		BICB #7,@ML.REG+40	
24085	104242	150077	106746			BISB R0,@ML.REG+40	
24086	104246	004767	111610			JSR PC,GD.BLK.XFER	
24087	104252	010100				MOV R1,R0	: LOOP,*
24088	104254	006300				ASL R0	1051
24089	104256	066007	104262			ADD 3\$(R0),PC	
24090	104262	000006			3\$:	.WORD 4\$-3\$	
24091	104264	000024				.WORD 5\$-3\$	
24092	104266	000044				.WORD 6\$-3\$	
24093	104270	152777	000004	106776	4\$:	BISB #4,@ML.REG+120	1051
24094	104276	142777	000002	106770		BICB #2,@ML.REG+120	1052
24095	104304	000416				BR 7\$	1052
24096	104306	142777	000004	106760	5\$:	BICB #4,@ML.REG+120	1052
24097	104314	142777	000002	106752		BICB #2,@ML.REG+120	1052
24098	104322	005002				CLR R2	1052
24099	104324	000410				BR 8\$	1051
24100	104326	152777	000004	106740	6\$:	BISB #4,@ML.REG+120	1053
24101	104334	152777	000002	106732		BISB #2,@ML.REG+120	1053
24102	104342	012702	000001		7\$:	MOV #1,R2	:*,EXPECTED
24103	104346	012777	000071	106600	8\$:	MOV #71,@ML.REG	1053
24104	104354	105777	106644		9\$:	TSTB @ML.REG+50	1053
24105	104360	100375				RPI 9\$	
24106	104362	010203				MOV R2,R3	: EXPECTED,*
24107	104364	005000				CLR R0	1054
24108	104366	005777	106642			TST @ML.REG+60	
24109	104372	100001				BPL 10\$	
24110	104374	005200				INC R0	
24111	104376	020003			10\$:	CMP R0,R3	
24112	104400	001446				BEQ 14\$	
24113	104402	020127	000001			CMP R1,#1	: LOOP,*

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:C

```

24115      ;ML4
24116      ;
24117
24118 104406 001405      BEQ      11$
24119 104410 104455      TRAP     55
24120 104412 000202      .WORD   202
24121 104414 010526      .WORD   SYNC
24122 104416 024052      .WORD   DUMPER
24123 104420 000404      BR      12$
24124 104422 104455      11$:   TRAP     55
24125 104424 000202      .WORD   202
24126 104426 010464      .WORD   ASYNC
24127 104430 024052      .WORD   DUMPER
24128 104432 012746 006406      MOV     #WORD.10,-(SP)      ;
24129 104436 012746 007472      MOV     #PHR.5,-(SP)      ;
24130 104442 012746 007302      MOV     #WORD.7,-(SP)      ;
24131 104446 012746 006034      MOV     #THR.FMT,-(SP)      ;
24132 104452 012746 000004      MOV     #4,-(SP)      ;
24133 104456 010600      MOV     SP,R0      ; SP,*
24134 104460 104414      TRAP     14      ;
24135 104462 005016      CLR     (SP)      ;
24136 104464 005777 106544      TST     @ML.REG+60      ;
24137 104470 100001      BPL     13$
24138 104472 005216      INC     (SP)
24139 104474 010246      13$:   MOV     R2,-(SP)      ; EXPECTED,*
24140 104476 012746 004222      MOV     #FMT.1,-(SP)
24141 104502 012746 000003      MOV     #3,-(SP)
24142 104506 010600      MOV     SP,R0      ; SP,*
24143 104510 104414      TRAP     14
24144 104512 062706 000020      ADD     #20,SP      ;
24145 104516 104467      14$:   TRAP     67      ;
24146 104520 006000      ROR     R0
24147 104522 103634      BLO     2$
24148 104524 005201      INC     R1      ; LOOP
24149 104526 020127 000002      CMP     R1,#2      ; LOOP,*
24150 104532 003630      BLE     2$
24151 104534 022626      CMP     (SP)+,(SP)+      ;
24152 104536 000207      RTS     PC
24153
24154
24155
24160
24161
24165

```

```

; Routine Size: 141 words
; Maximum stack depth per invocation: 14 words

```

24170
24171
24172
24173 104540
24174 104540 004767 177342
24175 104544 104466
24176 104546 006000
24177 104550 103773
24178 104552 000207
24179
24180
24181
24186
24187
24188 ;

;ML4
;
T59::
1\$: JSR PC,\$T59
TRAP 66
ROR R0
BLO 1\$
RTS PC

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

1055

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

10557 !<BLF/PAGE>

22-Dec-1980 09:24:31
22-Dec-1980 09:21:22

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>BL3ML4.BLI.2 (110)

```

24190 :ML4
24191 :
24192 :
24193 : 10558 !
24194 : 10559 BGNSTST;
24195 : 10560
24196 : 10561 !++
24197 : 10562 TEST NUMBER: TST 60
24198 : 10563
24199 : 10564 TEST NAME: PROM DATA TEST
24200 : 10565
24201 : 10566 TEST DESCRIPTION:
24202 : 10567
24203 : 10568 VERIFY THAT CHECK SUM VALUES FOR
24204 : 10569 ALL PRESENT MEMORY ARRAY UV PROM
24205 : 10570 LOCATIONS ARE ERROR FREE BY:
24206 : 10571
24207 : 10572 1. DOING MBUS TRANSFERS
24208 : 10573 AT ALL PRESENT BLOCKS
24209 : 10574 AND TESTING THE UNS BIT
24210 : 10575 TO BE CLEARED.
24211 : 10576
24212 : 10577 IMPLICIT INPUTS:
24213 : 10578
24214 : 10579 IO_BUF
24215 : 10580
24216 : 10581 A VECTOR OF 256 WORDS WHERE
24217 : 10582 DATA FOR MBUS READ AND WRITE
24218 : 10583 FUNCTIONS IS FOUND.
24219 : 10584
24220 : 10585
24221 : 10586 CHIP SIZ
24222 : 10587 INDICATED THE SIZE OF THE
24223 : 10588 ARRAY MODULES MOS RAMS.
24224 : 10589
24225 : 10590
24226 : 10591 !-
24227 : 10592
24228 : 10593 incr DSA_CNT from 0 to .LST_BLK do
24229 : 10594 begin
24230 : 10595 BGNSUB;
24231 : 10596 CLR_MBUS;
24232 : 10597 ECC_DIS = ONE;
24233 : 10598 MLWC = not 255;
24234 : 10599 MLBA = IO_BUF;
24235 : 10600 MLDA = .DSA_CNT;
24236 : 10601 MLCS1 = write;
24237 : 10602 TIME_OUT_LOOP;
24238 : 10603
24239 : 10604 if .UNS IS_SET
24240 : 10605 then
24241 : 10606 begin
24242 : 10607 ERRDF (107, ARR_DAT, DUMPER);
24243 : 10608 PRINTB (TWO_FMT, WRD 35, PHR_4);
24244 : 10609 PRINTB (FMT_9, .DSA_CNT);

```

!WRITE TO ALL PRESENT BLK'S AND CHECK UNS BIT

!DISABLE ECC
!LOAD WORD COUNT
!LOAD MBUS ADRS
!LOAD DSA
!DO A WRITE FUNCTION

!SEE IF XFERR CAUSED AN UNS ERROR

!ERROR IF SET

22-Dec-1980 09:24:31 TOPS-20 Bliss-16 V2(212)
22-Dec-1980 09:21:22 PA:<NEALE>BL3ML4.BLI.2 (110)

```

24246 :ML4
24247 :
24248 :
24249 :      10610
24250 :      10611      if .CHIP_SIZ eql 64      !NEED TO KNOW CHIP SIZE TO PRINT ARRAY NUMBER
24251 :      10612      then
24252 :      10613          begin
24253 :      10614          PRINTB (FMT_8, ((.DSA_CNT<11, 4>) + 1));      !64K MOS RAM
24254 :      10615          end
24255 :      10616      else
24256 :      10617          begin
24257 :      10618          PRINTB (FMT_8, ((.DSA_CNT<9, 4>) + 1));      !16K MOS RAM
24258 :      10619          end
24259 :      10620      end;
24260 :      10621      ENDSUB;
24261 :      10622      end;
24262 :      10623
24263 :      10624
24264 :      10625
24265 :      10626      ENDTST;
24269 :
24273 104554 004167 077264      $T60:      JSR      R1,$SAVE2      :
24274 104560 016702 105006      MOV      LST.BLK,R2      :
24275 104564 005001      CLR      R1      : DSA.CNT
24276 104566 000536      BR       6$      :
24277 104570 104402      1$:      TRAP     2      :
24278 104572 152777 000040 106414      BISB     #40,@ML.REG+40      :
24279 104600 016700 106776      MOV      ML.DUT,R0      :
24280 104604 042700 177770      BIC      #177770,R0      :
24281 104610 142777 000007 106376      BICB     #7,@ML.REG+40      :
24282 104616 150077 106372      BISB     R0,@ML.REG+40      :
24283 104622 152777 000002 106444      BISB     #2,@ML.REG+120      :
24284 104630 012777 177400 106326      MOV      #400,@ML.REG+10      :
24285 104636 012777 011600 106330      MOV      #10.BUF,@ML.REG+20      :
24286 104644 010177 106334      MOV      R1,@ML.REG+30      : DSA.CNT,*
24287 104650 012777 000061 106276      MOV      #61,@ML.REG      :
24288 104656 105777 106342      2$:      TSTB     @ML.REG+50      :
24289 104662 100375      BPL      2$      :
24290 104664 032777 040000 106342      BIT      #40000,@ML.REG+60      :
24291 104672 001470      BIC      5$      :
24292 104674 104455      TRAP     5$      :
24293 104676 000153      .WORD    153      :
24294 104700 010570      .WORD    ARR.DAT      :
24295 104702 024052      .WORD    DUMPER      :
24296 104704 012746 007454      MOV      #PHR.4,-(SP)      :
24297 104710 012746 006666      MOV      #WRD.35,-(SP)      :
24298 104714 012746 006024      MOV      #TWO.FMT,-(SP)      :
24299 104720 012746 000003      MOV      #3,-(SP)

```

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA: <

```

24301          :ML4
24302          :
24303          :
24304 104724 010600      MOV    SP,R0          : SP,*
24305 104726 104414      TRAP   14
24306 104730 010116      MOV    R1,(SP)       : DSA.CNT,*
24307 104732 012746 004644  MOV    #FMT.9,-(SP)
24308 104736 012746 000002  MOV    #2,-(SP)
24309 104742 010600      MOV    SP,R0          : SP,*
24310 104744 104414      TRAP   14
24311 104746 026727 104616 000100  CMP    CHIP.SIZ,#100
24312 104754 001020      BNE   3$
24313 104756 010100      MOV    R1,R0         : DSA.CNT,*
24314 104760 006200      ASR   R0
24315 104762 006200      ASR   R0
24316 104764 006200      ASR   R0
24317 104766 000300      SWAB  R0
24318 104770 042700 177760      BIC   #177760,R0
24319 104774 010046      MOV    R0,-(SP)
24320 104776 005216      INC   (SP)
24321 105000 012746 004610  MOV    #FMT.8,-(SP)
24322 105004 012746 000002  MOV    #2,-(SP)
24323 105010 010600      MOV    SP,R0          : SP,*
24324 105012 104414      TRAP   14
24325 105014 000415      BR    4$
24326 105016 010100      3$:  MOV    R1,R0         : DSA.CNT,*
24327 105020 006200      ASR   R0
24328 105022 000300      SWAB  R0
24329 105024 042700 177760      BIC   #177760,R0
24330 105030 010046      MOV    R0,-(SP)
24331 105032 005216      INC   (SP)
24332 105034 012746 004610  MOV    #FMT.8,-(SP)
24333 105040 012746 000002  MOV    #2,-(SP)
24334 105044 010600      MOV    SP,R0          : SP,*
24335 105046 104414      TRAP   14
24336 105050 062706 000022  4$:  ADI   #22,SP
24337 105054 104467      5$:  TRAP   67
24338 105056 006000      ROR   R0
24339 105060 103643      BLO   1$
24340 105062 005201      INC   R1              : DSA.CNT
24341 105064 020102      6$:  CMP    R1,R2         : DSA.CNT,*
24342 105066 003640      BLE   1$
24343 105070 000207      RTS    PC
24344          :
24345          :
24346          :
24351          :
24352          :

```

: Routine Size: 103 words
: Maximum stack depth per invocation: 12 words

24357
24358
24359
24360
24364
24365
24366
24367
24368
24369
24370
24371
24372
24377
24378
24379

105072
105072 004767 177456
105076 104466
105100 006000
105102 103773
105104 000207

:ML4
:

T60::
1\$:

JSR PC,\$T60
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

10627 !<BLF/PAGE>

22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

1062

```
24381 :ML4
24382 :
24383 :
24384 : 10628 ! THE CLEANUP CODING SECTION IS EXECUTED AFTER THE
24385 : 10629 ! HARDWARE TESTS ARE RUN ON A LOGICAL UNIT.
24386 : 10630 BGNCLN;
24387 : 10631 CLR = ONE; !CLEAR THE MASS BUS
24388 : 10632 return;
24389 : 10633 ENDCLN;
24393 :
24397 105106 152777 000040 106100 LCLEAN: BISB #40,@ML.REG+40 ; 1063
24398 105114 000207 RTS PC ; 1062
24399 :
24400 : ; Routine Size: 4 words
24401 : ; Maximum stack depth per invocation: 0 words
24406 :
24407 :
24411 :
24415 105116 LSCLEAN::
24416 105116 004767 177764 JSR PC,LCLEAN ; 1063
24417 105122 104412 TRAP 12
24418 105124 000207 RTS PC
24419 :
24420 : ; Routine Size: 4 words
24421 : ; Maximum stack depth per invocation: 0 words
24426 :
24427 :
24428 : 10634 LASTAD;
24429 : 10635 BGNSETUP (0);
24430 : 10636 ENDSETUP;
```


22-Dec-1980 09:24:31 TOPS
22-Dec-1980 09:21:22 PA:<

```
24435 ;ML4
24436 ;
24437
24438
24439 105126 105132 BLSLAS::.WORD TSFREE
24440 105130 000000 .WORD <<TSFREE-<BLSLAS+4>>/2>
24441 105132 000000 TSFREE::.WORD 0
24442
24443
24444 105132 LSLAST== BLSLAS+4
24445 000000 TSPTHV== 0
24446
24447
24451 105134 SEND.LINK::
24452 105134 000207 RTS PC ;
24453
24454 ; Routine Size: 1 word
24455 ; Maximum stack depth per invocation: 0 words
24460
24461
24462 : 10637 end
24463 : 10638
24464 : 10639 eludom
24468
24469 ; OTS external references
24470 .GLOBL BLSGT2, $SAVE5, $SAVE4, $SAVE3
24471 .GLOBL $SAVE2, BL$PU2, BL$PU1, BLSGT1
24472 .GLOBL BL$DIV, BL$MOD, BL$MUL
24473
24474
24475
24476
24477
24478
24479 ; Size: 14698 code + 1916 data words
24480 ; Run Time: 03:04.1
24481 ; Elapsed Time: 03:48.6
24482 ; Memory Used: 118 pages
24483 ; Compilation Complete
24484
24485 000001 .END
```

1063

ADR = 000020 G	C\$CLP1= 000006	EF.RES= 000037 G	FNC.3 = 007750	IER = 020000 G
ARR.DA= 010570	C\$CVEC= 000036	EF.STA= 000040 G	FNC.4 = 007756	INTER = 010672
ARR.IN 011562	C\$DCLN= 000044	EIG.FM= 006142	FNC.5 = 007774	IO.BUF 011600
ARR.16 011574	C\$DODU= 000051	ELV.FM= 006244	FNC.6 = 010004	ISR = 000100 G
ASSEMB= 000010	C\$DRPT= 000024	ERRBLK 002160 G	FNC.7 = 010012	IXE = 004000 G
ASYNCR = 010464	C\$SDU = 000053	ERRMSG 002156 G	FNC.8 = 010022	ISAU = 000041
A.CAL 013140	C\$EDIT= 000003	ERRNER 002154 G	FNC.9 = 010036	ISAUTO= 000041
A.GEN 013146	C\$ERPF= 000055	ERRTYP 002152 G	FORCE. 015602	ISCLN = 000041
BIT0 = 000001 G	C\$ERHR= 000056	ERR.CH 015212	FOR.EC 015760	ISDU = 000041
BIT00 = 000001 G	C\$ERRO= 000060	EVL = 000004 G	FOR.FM= 006046	ISHRD = 000041
BIT01 = 000002 G	C\$ERCF= 000054	ESEND = 002100	FSAU = 000015	ISINIT= 000041
BIT02 = 000004 G	C\$ERSO= 000057	ESLOAD= 000035	FSAUTO= 000020	ISMOD = 000041
BIT03 = 000010 G	C\$ESCA= 000010	E2.TEM= 011156	F\$BGN = 000040	ISMSG = 000041
BIT04 = 000020 G	C\$ESEG= 000005	FIND.C 015430	F\$CLEA= 000007	ISPROT= 000040
BIT05 = 000040 G	C\$ESUB= 000003	FIRST. 016040	F\$DU = 000016	ISPTAB= 000041
BIT06 = 000100 G	C\$ETST= 000001	FIV.FM= 006062	F\$END = 000041	ISPRR = 000041
BIT07 = 000200 G	C\$EXIT= 000032	FMT.1 = 004222	F\$HARD= 000004	ISRPT = 000041
BIT08 = 000400 G	C\$GETB= 000026	FMT.10= 004676	F\$HW = 000013	ISSEG = 000041
BIT09 = 001000 G	C\$GETW= 000027	FMT.11= 004750	F\$INIT= 000006	ISSETU= 000041
BIT1 = 000002 G	C\$GMAN= 000043	FMT.12= 005004	F\$JMP = 000050	ISSFT = 000041
BIT10 = 002000 G	C\$GPHR= 000042	FMT.13= 005034	F\$MOD = 000000	ISSRV = 000041
BIT11 = 004000 G	C\$GPLO= 000030	FMT.14= 005110	F\$MSG = 000011	ISSUB = 000041
BIT12 = 010000 G	C\$GPRI= 000040	FMT.15= 005150	F\$PROT= 000021	ISTST = 000041
BIT13 = 020000 G	C\$INIT= 000011	FMT.16= 005216	F\$PWR = 000017	JSJMP = 000167
BIT14 = 040000 G	C\$INLP= 000020	FMT.17= 005306	F\$RPT = 000012	LAST.B 016114
BIT15 = 100000 G	C\$MANI= 000050	FMT.18= 005342	F\$SEG = 000003	LAU 004210
BIT2 = 000004 G	C\$MEM = 000031	FMT.19= 005412	F\$SOFT= 000005	LAUTO 004164
BIT3 = 000010 G	C\$MSG = 000023	FMT.2 = 004266	F\$SRV = 000010	LCLEAN 105106
BIT4 = 000020 G	C\$OPEN= 000034	FMT.20= 005462	F\$SUB = 000002	LDU 004176
BIT5 = 000040 G	C\$PNTB= 000014	FMT.21= 005534	F\$SW = 000014	LD.LNG 017124
BIT6 = 000100 G	C\$PNTF= 000017	FMT.22= 005606	F\$TEST= 000001	LINIT 024714
BIT7 = 000200 G	C\$PNTS= 000016	FMT.23= 005652	GD.BLK 016062	LOAD.S 013604
BIT8 = 000400 G	C\$PNTX= 000015	FMT.24= 005722	GOOD.B 011564	LOE = 040000 G
BIT9 = 001000 G	C\$QIO = 000377	FMT.25= 005746	G\$CNTD= 000200	LOT = 000010 G
BL\$DIV 003752 G	C\$RDBU= 000007	FMT.26= 005764	G\$DELM= 000372	LRPT 004152
BL\$GT1 003016 G	C\$REFG= 000047	FMT.3 = 004354	G\$DISP= 000003	LST.AR 011576
BL\$GT2 003140 G	C\$RESE= 000033	FMT.4 = 004400	G\$EXCP= 000400	LST.BL 011572
BL\$LAS 105126 G	C\$REVI= 000003	FMT.5 = 004430	G\$HILI= 000002	LST.DU 013134
BL\$MOD 003764 G	C\$RFLA= 000021	FMT.6 = 004532	G\$LOLI= 000001	LSACP 002110 G
BL\$MUL 003526 G	C\$RPT = 000025	FMT.7 = 004562	G\$NO = 000000	LSAPT 002036 G
BL\$PU1 003302 G	C\$SEFG= 000046	FMT.8 = 004610	G\$OFFS= 000400	LSAU 004212 G
BL\$PU2 003376 G	C\$SPRI= 000041	FMT.9 = 004644	G\$OF SI= 000376	LSAUT 002070 G
BL\$SHF 003776 G	C\$SVEC= 000037	FNC.1 = 007726	G\$PRMA= 000001	LSAUTO 004166 G
BOE = 000400 G	C\$TPRI= 000013	FNC.10= 010050	G\$PRMD= 000002	L\$CCP 002106 G
B.CAL 013142	DATA.L= 010732	FNC.11= 010060	G\$PRML= 000000	L\$CLEA 105116 G
B.GEN 013150	DAT.DM 014162	FNC.12= 010100	G\$RADA= 000140	L\$CO 002032 G
CAL.CR 014612	DFPTBL 002356 G	FNC.13= 010112	G\$RADB= 000000	L\$DEPO 002011 G
CHIP.S 011570	DIAGMC= 000000	FNC.14= 010122	G\$RADD= 000040	L\$DESC 002130 G
C\$AU = 000052	DIVMOD 003570	FNC.15= 010136	G\$RADL= 000120	L\$DESP 002076 G
C\$AUTO= 000061	DRIVE. 013132	FNC.16= 010150	G\$RADO= 000020	L\$DEVP 002060 G
C\$BRK = 000022	DT.1 013534	FNC.17= 010162	G\$XFER= 000004	L\$DISP 002164 G
C\$BSEG= 000004	DUMPER 024052 G	FNC.18= 010172	G\$YES = 000010	L\$DLY 002116 G
C\$BSUB= 000002	D1.TEM= 011152	FNC.19= 010204	HELP = 000000	L\$DTP 002040 G
C\$CEFG= 000045	D2.TEM= 011154	FNC.2 = 007742	HOE = 100000 G	L\$DTYP 002034 G
C\$CLCK= 000062	EF.CON= 000036 G	FNC.21= 010216	HW.OR. 011160	L\$DU 004200 G
C\$CLEA= 000012	EF.NEW= 000035 G	FNC.22= 010230	IBE = 010000 G	L\$DUT 002072 G
C\$CLOS= 000035	EF.PWR= 000034 G	FNC.23= 010242	IDU = 000040 G	L\$DVTY 002122 G

LSEF	002052	G	OSAPTS=	000001	P.AAT	005462	P.ACY	006732	P.AFD	010012
LSEVI	002044	G	OSAU =	000001	P.AAU	005534	P.ACZ	006746	P.AFE	010022
LSEVRT	002152	G	OSBGNR=	000001	P.AAV	005606	P.ADA	006754	P.AFF	010036
LSETP	002102	G	OSBGNS=	000001	P.AAW	005652	P.ADB	006764	P.AFG	010050
LSEXP1	002046	G	OSDU =	000001	P.AAX	005722	P.ADC	006772	P.AFH	010060
LSEXP4	002064	G	OSERRT=	000001	P.AAY	005746	P.ADD	007004	P.AFI	010100
LSEXP5	002066	G	OSGNSW=	000001	P.AAZ	005764	P.ADE	007016	P.AFJ	010112
LSHARD	002402	G	OSPOIN=	000001	P.ABA	006016	P.ADF	007026	P.AFK	010122
LSHIME	002120	G	OSSETU=	000001	P.ABB	006024	P.ADG	007036	P.AFL	010136
LSHPCP	002016	G	PAR.DI	011566	P.ABC	006034	P.ADH	007046	P.AFM	010150
LSHPTP	002022	G	PD.TEM	013120	P.ABD	006046	P.ADI	007054	P.AFN	010162
LSHW	002356	G	PHR.1 =	007366	P.ABE	006062	P.ADJ	007066	P.AFO	010172
LSICP	002104	G	PHR.10=	007564	P.ABF	006100	P.ADK	007074	P.AFP	010204
LSINIT	025660	G	PHR.11=	007576	P.ABG	006120	P.ADL	007104	P.AFQ	010216
LSLADP	002026	G	PHR.12=	007614	P.ABH	006142	P.ADM	007114	P.AFR	010230
LSLAST=	105132	G	PHR.13=	007636	P.ABI	006166	P.ADN	007122	P.AFS	010242
LSLOAD	002100	G	PHR.14=	007662	P.ABJ	006214	P.ADO	007130	P.AFT	010256
LSLUN	002074	G	PHR.15=	007714	P.ABK	006244	P.ADP	007140	P.AFU	010266
LSPREV	002050	G	PHR.2 =	007404	P.ABL	006276	P.ADQ	007152	P.AFV	010274
LSNAME	002000	G	PHR.3 =	007422	P.ABM	006302	P.ADR	007156	P.AFW	010302
LSPRIO	002042	G	PHR.4 =	007454	P.ABN	006314	P.ADS	007174	P.AFX	010310
LSPROT	002750	G	PHR.5 =	007472	P.ABO	006322	P.ADT	007210	P.AFY	010316
LSPRT	002112	G	PHR.6 =	007504	P.ABP	006330	P.ADU	007216	P.AFZ	010324
LSREPP	002062	G	PHR.7 =	007516	P.ABQ	006336	P.ADV	007224	P.AGA	010332
LSREV	002010	G	PHR.8 =	007540	P.ABR	006344	P.ADW	007232	P.AGB	010340
LSRPT	004154	G	PHR.9 =	007552	P.ABS	006360	P.ADX	007240	P.AGC	010346
LSSOFT	002720	G	PNT =	001000	P.ABT	006374	P.ADY	007246	P.AGD	010354
LSSPC	002056	G	PRI =	002000	P.ABU	006406	P.ADZ	007252	P.AGE	010362
LSSPCP	002020	G	PRI00 =	000000	P.ABV	006416	P.AEA	007260	P.AGF	010370
LSSPTP	002024	G	PRI01 =	000040	P.ABW	006426	P.AEB	007272	P.AGG	010376
LSSTA	002030	G	PRI02 =	000100	P.ABX	006436	P.AEC	007302	P.AGH	010404
LSSW	002376	G	PRI03 =	000140	P.ABY	006442	P.AED	007310	P.AGI	010412
LSTEST	002114	G	PRI04 =	000200	P.ABZ	006454	P.AEE	007316	P.AGJ	010420
LSTIML	002014	G	PRI05 =	000240	P.ACA	006462	P.AEF	007330	P.AGK	010430
LSUNIT	002012	G	PRI06 =	000300	P.ACB	006470	P.AEG	007340	P.AGL	010436
L10000	002374	G	PRI07 =	000340	P.ACC	006502	P.AEH	007346	P.AGM	010444
L10001	002400	G	PRSN	002376	P.ACD	006506	P.AEI	007366	P.AGN	010454
L10002	002476	G	PTBL.P	011556	P.ACE	006514	P.AEJ	007404	P.AGO	010464
L10003	002726	G	P.AAA	004222	P.ACF	006522	P.AEK	007422	P.AGP	010526
MEM.AR=	010630	G	P.AAB	004266	P.ACG	006530	P.AEL	007454	P.AGQ	010570
ML.DUT	013602	G	P.AAC	004354	P.ACH	006544	P.AEM	007472	P.AGR	010630
ML.LUN	013600	G	P.AAD	004400	P.ACI	006552	P.AEN	007504	P.AGS	010672
ML.REG	013154	G	P.AAE	004430	P.ACJ	006560	P.AEO	007516	P.AGT	010732
MSGH1	002476	G	P.AAF	004532	P.ACK	006574	P.AEP	007540	P.AGU	010772
MSGH2	002512	G	P.AAG	004562	P.ACL	006602	P.AEQ	007552	P.AGV	011026
MSGH3	002540	G	P.AAH	004610	P.ACM	006610	P.AER	007564	P.AGW	011060
MSGH4	002563	G	P.AAI	004644	P.ACN	006624	P.AES	007576	P.AGX	011106
MSGH5	002614	G	P.AAJ	004676	P.ACO	006632	P.AET	007614	P.CAL	013144
MSGH6	002646	G	P.AAK	004750	P.ACP	006646	P.AEU	007636	P.GEN	013152
MSGH7	002672	G	P.AAL	005004	P.ACQ	006654	P.AEV	007662	RAS.IN	013124
MSGS1	002726	G	P.AAM	005034	P.ACR	006660	P.AEW	007714	RD.CS1	017532
MSDUMP	024062	G	P.AAN	005110	P.ACS	006666	P.AEX	007726	RD.DA	020066
NIB.SA	011152	G	P.AAO	005150	P.ACT	006674	P.AEY	007742	RD.DAT	013130
NIN.FM=	006166	G	P.AAP	005216	P.ACU	006700	P.AEZ	007750	RD.DS	022766
ONEFIL=	000001	G	P.AAQ	005306	P.ACV	006710	P.AFA	007756	RD.D1	021654
ONE.FM=	006016	G	P.AAR	005342	P.ACW	006716	P.AFB	007774	RD.D2	022166
OP.NUM	011560	G	P.AAS	005412	P.ACX	006726	P.AFC	010004	RD.D3	022514

RD.EE	021530	TST.LN	016146	T33	060512 G	WRD.30=	006624	WRT.D2	022102
RD.EL	021466	TWO.FM=	006024	T34	061446 G	WRD.31=	006632	WRT.D3	022414
RD.ER	017710	TSARGC=	000003	T35	062100 G	WRD.32=	006646	WRT.EE	021526
RD.E1	020644	TSCODE=	000130	T36	063136 G	WRD.33=	006654	WRT.EL	021464
RD.E2	021064	TSERRN=	000000	T37	064044 G	WRD.34=	006660	WRT.ER	017640
RD.MR	020244	TSEXCP=	000000	T38	065050 G	WRD.35=	006666	WRT.E1	020560
RD.PA	020436	TSFREE	105132 G	T39	065476 G	WRD.36=	006674	WRT.E2	020766
RD.PD	021326	TSGMAN=	000000	T4	027222 G	WRD.37=	006700	WRT.MR	020174
RD.REG	023462	TSHILI=	000007	T40	066464 G	WRD.38=	006710	WRT.PA	020352
REG.IN	013136	TSLAST=	000000	T41	067516 G	WRD.39=	006716	WRT.PD	021220
REG.1 =	010256	TSLOLI=	000000	T42	070270 G	WRD.4 =	006322	WRT.RE	023074
REG.10=	010346	TLSYM=	010000	T43	071224 G	WRD.40=	006726	WT.DAT	013126
REG.11=	010354	TSNEST=	177777	T44	071434 G	WRD.41=	006732	W.C.SI	013122
REG.12=	010362	TSNS0 =	000000	T45	071620 G	WRD.42=	006746	XOR.LN	016642
REG.13=	010370	TSNS1 =	000021	T46	072316 G	WRD.43=	006754	XSALWA=	000000
REG.14=	010376	TSPTHV=	000000 G	T47	072610 G	WRD.44=	006764	XSALS=	000040
REG.15=	010404	TSPTNU=	000000	T48	073502 G	WRD.45=	006772	XSOFFS=	000400
REG.16=	010412	TSSAVL=	177777	T49	074206 G	WRD.46=	007004	XSTRUE=	000020
REG.17=	010420	TSSEGL=	177777	T5	027644 G	WRD.47=	007016	SEND.L	105134 G
REG.18=	010430	TSSUBN=	000000	T50	074624 G	WRD.48=	007026	\$PATCH	002756 G
REG.19=	010436	TSTAGL=	177777	T51	075676 G	WRD.49=	007036	\$SAVE2	004044 G
REG.2 =	010266	TSTAGN=	010005	T52	076450 G	WRD.5 =	006330	\$SAVE3	004060 G
REG.20=	010444	TSTEMP=	000000	T53	077250 G	WRD.50=	007046	\$SAVE4	004076 G
REG.21=	010454	TSTEST=	000000	T54	100272 G	WRD.51=	007054	\$SAVE5	004116 G
REG.3 =	010274	TSTSTM=	177777	T55	101142 G	WRD.52=	007066	\$T1	025670
REG.4 =	010302	TSTSTS=	000000	T56	102244 G	WRD.53=	007074	\$T10	032366
REG.5 =	010310	TSSHAR=	010002	T57	103466 G	WRD.54=	007104	\$T11	033030
REG.6 =	010316	TSSHW =	010000	T58	104072 G	WRD.55=	007114	\$T12	033274
REG.7 =	010324	TSSPRO=	010004	T59	104540 G	WRD.56=	007122	\$T13	033704
REG.8 =	010332	TSSSOF =	010003	T6	030302 G	WRD.57=	007130	\$T14	035056
REG.9 =	010340	TSSSW =	010001	T60	105072 G	WRD.58=	007140	\$T15	036324
REM.TB	010434	T1	026056 G	T7	031132 G	WRD.59=	007152	\$T16	037572
RE2	004144	T10	033014 G	T8	032116 G	WRD.6 =	006336	\$T17	040472
RE3	004142	T11	033260 G	T9	032352 G	WRD.60=	007156	\$T18	041666
RE4	004140	T12	033670 G	UAM	= 000200 G	WRD.61=	007174	\$T19	042124
RH.ADD	013572	T13	035042 G	WRD.1 =	006276	WRD.62=	007210	\$T2	026072
RH.ERR=	011060	T14	036310 G	WRD.10=	006406	WRD.63=	007216	\$T20	043274
RH.TYP	013574	T15	037556 G	WRD.11=	006416	WRD.64=	007224	\$T21	044234
RH.VEC	013576	T16	040456 G	WRD.12=	006426	WRD.65=	007232	\$T22	044630
SC.SET=	010772	T17	041652 G	WRD.13=	006436	WRD.67=	007240	\$T23	045276
SEV.FM=	006120	T18	042110 G	WRD.14=	006442	WRD.68=	007246	\$T24	045634
SFPTBL	002376 G	T19	043260 G	WRD.15=	006454	WRD.69=	007252	\$T25	047010
SIX.FM=	006100	T2	026322 G	WRD.16=	006462	WRD.7 =	006344	\$T26	047566
STACK	012612	T20	044220 G	WRD.17=	006470	WRD.70=	007260	\$T27	050712
STK.OF	012600	T21	044614 G	WRD.18=	006502	WRD.71=	007272	\$T28	051470
STRIPP	014214	T22	045262 G	WRD.19=	006506	WRD.72=	007302	\$T29	052454
SVCGBL=	177777	T23	045620 G	WRD.2 =	006302	WRD.73=	007310	\$T3	026336
SVCINS=	177777	T24	046774 G	WRD.20=	006514	WRD.74=	007316	\$T30	053566
SVCSUB=	177777	T25	047552 G	WRD.21=	006522	WRD.75=	007330	\$T31	054362
SVCTAG=	177777	T26	050676 G	WRD.22=	006530	WRD.76=	007340	\$T32	055756
SVCTST=	177777	T27	051454 G	WRD.23=	006544	WRD.77=	007346	\$T33	057514
SYNC =	010526	T28	052440 G	WRD.24=	006552	WRD.8 =	006360	\$T34	060526
SLSYM=	010000	T29	053552 G	WRD.25=	006560	WRD.9 =	006374	\$T35	061462
TEN.FM=	006214	T3	026530 G	WRD.26=	006574	WRT.CS	017462	\$T36	062114
THR.FM=	006034	T30	054346 G	WRD.27=	006602	WRT.DA	020016	\$T37	063152
TIME.O=	011106	T31	055742 G	WRD.29=	006610	WRT.DS	022764	\$T38	064060
TRELE.=	011026	T32	057500 G	WRD.3 =	006314	WRT.D1	021570	\$T39	065064

\$T4	026544	\$T45	071450	\$T50	074222	\$T55	100306	\$T6	027660
\$T40	065512	\$T46	071634	\$T51	074640	\$T56	101156	\$T60	104554
\$T41	066500	\$T47	072332	\$T52	075712	\$T57	102260	\$T7	030316
\$T42	067532	\$T48	072624	\$T53	076464	\$T58	103502	\$T8	031146
\$T43	070304	\$T49	073516	\$T54	077264	\$T59	104106	\$T9	032132
\$T44	071240	\$T5	027236						

. ABS. 105136 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 34695 WORDS (136 PAGES)

DYNAMIC MEMORY: 21558 WORDS (82 PAGES)

ELAPSED TIME: 00:11:10

ML11,ML11/-SP/CR:SYM=SVC/ML,CZMLAB.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	#84-2189
ARR.DA	=	010570	#88-2389 204-8071 209-8320 215-8617 223-9002 271-11447 272-11511 273-11567 309-13418
			315-13685 321-14034 322-14055 322-14076 323-14097 323-14114 334-14717 390-17656 396-17972
			419-19218 519-24294
ARR.IN	=	011562	#79-1880 *186-7192 *186-7212 289-12395 289-12436 290-12442 332-14603 342-15126 342-15135
			361-16136 362-16154 364-16301 365-16319 412-18824 412-18828
ARR.16	=	011574	#79-1887 *186-7182 *186-7193 *186-7213 *186-7214 289-12437
ASSEMB	=	000010	4-13 4-13
ASYN	=	010464	#88-2387 196-7647 200-7860 204-8053 209-8302 215-8597 216-8659 222-8989 229-9334
			230-9358 230-9378 233-9527 236-9704 237-9725 237-9742 241-9954 242-9973 242-9990
			242-10007 243-10028 243-10063 244-10086 244-10110 248-10330 249-10349 249-10366 249-10383
			250-10404 250-10439 251-10462 251-10486 256-10735 257-10754 257-10771 257-10788 258-10809
			258-10844 259-10867 259-10891 264-11107 264-11123 264-11142 265-11163 265-11181 265-11199
			277-11765 281-11997 282-12018 282-12036 283-12072 283-12105 284-12151 288-12344 288-12360
			288-12377 289-12410 290-12456 293-12589 293-12605 293-12621 294-12641 297-12800 298-12823
			298-12844 302-13025 328-14399 347-15410 355-15842 366-16403 413-18898 429-19737 430-19761
			431-19804 431-19827 431-19843 434-19988 496-23137 505-23579 516-24126
A.CAL	=	013140	#79-1904
A.GEN	=	013146	#79-1907 98-2891 *98-2892 *98-2893 101-3015 *453-21006 454-21036 *458-21249 459-21279
BIT0	=	000001 G	#84-2173
BIT00	=	000001 G	#84-2163
BIT01	=	000002 G	#84-2162
BIT02	=	000004 G	#84-2161
BIT03	=	000010 G	#84-2160
BIT04	=	000020 G	#84-2159
BIT05	=	000040 G	#84-2158
BIT06	=	000100 G	#84-2157
BIT07	=	000200 G	#84-2156
BIT08	=	000400 G	#84-2155
BIT09	=	001000 G	#84-2154
BIT1	=	000002 G	#84-2172
BIT10	=	002000 G	#84-2153
BIT11	=	004000 G	#84-2152
BIT12	=	010000 G	#84-2151
BIT13	=	020000 G	#84-2150
BIT14	=	040000 G	#84-2149
BIT15	=	100000 G	#84-2148
BIT2	=	000004 G	#84-2171
BIT3	=	000010 G	#84-2170
BIT4	=	000020 G	#84-2169
BIT5	=	000040 G	#84-2168
BIT6	=	000100 G	#84-2167
BIT7	=	000200 G	#84-2166
BIT8	=	000400 G	#84-2165
BIT9	=	001000 G	#84-2164
BLSDIV	=	003752 G	#37-274 102-3086 483-22457 489-22772 523-24472
BLSGT1	=	003016 G	#9-143 396-17964 523-24471
BLSGT2	=	003140 G	#15-153 94-2707 98-2863 103-3112 281-11991 283-12066 284-12145 307-13305 309-13388
			321-13988 334-14689 341-15084 347-15383 353-15715 355-15828 362-16179 365-16343 372-16699
			379-17045 389-17624 390-17637 419-19205 448-20756 483-22481 490-22822 523-24470
BLSLAS	=	105126 G	#523-24439 523-24440 523-24444
BLSMOD	=	003764 G	#38-304 103-3094 483-22496 489-22776 523-24472

SYMBOL	VALUE	CROSS REFERENCE	REFERENCES
BL\$PUL	003526	G	#34-128 495-23053 523-24472
BL\$PU1	003302	G	#21-140 453-20991 523-24471
BL\$PU2	003376	G	#28-183 101-2990 101-2999 101-3008 283-12099 390-17671 484-22513 490-22793 523-24471
BL\$SHF	003776	G	9-155 9-158 9-162 9-166 10-191 16-175 16-178 16-182 16-186 16-207 21-143 21-147 21-153 22-180 28-192 28-196 28-202 29-234 #39-338
BOE	= 000400	G	#84-2193
B.CAL	013142		#79-1905 98-2897 *98-2898 *98-2899 101-3017 *453-21007 454-21037 *458-21250 459-21280
B.GEN	013150		#79-1908 101-3000 101-3017 *453-21004 454-21030 *458-21247 458-21269
CAL.CR	014612		#97-2832 453-21009 458-21252
CHIP.S	011570		#79-1884 *186-7191 *186-7211 520-24311
C\$AU	= 000052		#4-13
C\$AUTO	= 000061		#4-13
C\$BRK	= 000022		#4-13
C\$BSEG	= 000004		#4-13
C\$BSUB	= 000002		#4-13
C\$CEFG	= 000045		#4-13
C\$CLCK	= 000062		#4-13
C\$CLEA	= 000012		#4-13
C\$CLOS	= 000035		#4-13
C\$CLP1	= 000006		#4-13
C\$VEC	= 000036		#4-13
C\$DCLN	= 000044		#4-13
C\$DODU	= 000051		#4-13
C\$DRPT	= 000024		#4-13
C\$DU	= 000053		#4-13
C\$EDIT	= 000003		#4-13
C\$ERDF	= 000055		#4-13
C\$ERHR	= 000056		#4-13
C\$ERRO	= 000060		#4-13
C\$ERSF	= 000054		#4-13
C\$ERSO	= 000057		#4-13
C\$ESCA	= 000010		#4-13
C\$ESEG	= 000005		#4-13
C\$ESUB	= 000003		#4-13
C\$ETST	= 000001		#4-13
C\$EXIT	= 000032		#4-13
C\$GETB	= 000026		#4-13
C\$GETW	= 000027		#4-13
C\$GMAN	= 000043		#4-13
C\$GPHR	= 000042		#4-13
C\$GPLO	= 000030		#4-13
C\$GPRI	= 000040		#4-13
C\$INIT	= 000011		#4-13
C\$INLP	= 000020		#4-13
C\$MANI	= 000050		#4-13
C\$MEM	= 000031		#4-13
C\$MSG	= 000023		#4-13
C\$OPEN	= 000034		#4-13
C\$PNTB	= 000014		#4-13
C\$PNTF	= 000017		#4-13
C\$PNTS	= 000016		#4-13

4-65

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	CREF
CS&PNTX	=	000015	#4-13	
CS&QIO	=	000377	#4-13	
CS&RDBU	=	000007	#4-13	
CS&REFG	=	000047	#4-13	
CS&RESE	=	000033	#4-13 #4-13	
CS&REVI	=	000003	#4-13 4-65	
CS&RFLA	=	000021	#4-13	
CS&RPT	=	000025	#4-13	
CS&SEFG	=	000046	#4-13	
CS&SPRI	=	000041	#4-13	
CS&SVEC	=	000037	#4-13	
CS&TPRI	=	000013	#4-13	
DATA.L	=	010732	#88-2392	
DAT.DM		014162	#91-2556 93-2663 321-14015 346-15321 346-15342 370-16618 371-16658 377-16968 378-17004	
			389-17585 417-19120 418-19161 483-22459 490-22800	
DFPTBL	=	002356	G #4-119	
DIAGMC	=	000000	G 4-13 4-13	
DIVMOD		003570	#35-196 37-275 38-305	
DRIVE.		013132	#79-1898 *186-7188 *186-7208 216-8655 216-8668	
DT.1		013534	#83-2117 495-23055 495-23057 495-23073 497-23182	
DUMPER		024052	G #179-6800 190-7359 193-7497 196-7648 199-7802 200-7861 204-8054 204-8063 204-8072	
			209-8303 209-8312 210-8325 215-8598 215-8609 215-8618 216-8660 222-8990 223-9003	
			229-9335 230-9359 230-9379 233-9528 236-9705 237-9726 237-9743 241-9955 242-9974	
			242-9991 242-10008 243-10029 243-10064 243-10069 244-10087 244-10111 244-10116 248-10331	
			249-10350 249-10367 249-10384 250-10405 250-10440 250-10445 251-10463 251-10487 251-10492	
			252-10517 256-10736 257-10755 257-10772 257-10789 258-10810 258-10845 258-10850 259-10868	
			259-10892 259-10897 260-10922 264-11108 264-11124 264-11143 265-11164 265-11182 265-11200	
			271-11448 272-11512 273-11588 277-11766 282-12002 282-12019 282-12037 283-12073 283-12106	
			284-12152 288-12345 288-12361 288-12378 289-12411 290-12457 293-12590 293-12606 293-12622	
			294-12642 298-12805 298-12824 298-12845 302-13026 302-13041 308-13346 309-13419 315-13686	
			321-14035 322-14056 322-14077 323-14098 323-14115 328-14400 334-14718 342-15141 347-15411	
			353-15730 355-15843 364-16281 366-16404 372-16732 373-16758 379-17059 383-17274 390-17651	
			396-17973 401-18265 402-18301 403-18340 407-18574 412-18834 413-18999 419-19219 425-19479	
			425-19501 425-19520 429-19738 430-19762 431-19805 431-19828 431-19844 434-19989 437-20110	
			440-20282 440-20317 443-20482 449-20775 454-21020 458-21259 464-21521 464-21544 465-21611	
			466-21641 471-21896 471-21920 477-22186 477-22210 484-22552 491-22853 496-23138 497-23157	
			498-23212 504-23525 505-23580 507-23684 511-23885 516-24122 516-24127 519-24295	
D1.TEM	=	011152	#88-2397 *94-2691 *328-14368 *333-14671 340-15015 *341-15069 *347-15369 *353-15701 *355-15814	
			*371-16681 *378-17027 *389-17607 *419-19191 *448-20748 *484-22514 484-22520 *489-22767 490-22797	
D2.TEM	=	011154	#88-2398 *94-2692 *328-14369 *333-14672 340-15016 *341-15070 *347-15370 *353-15702 *355-15815	
			*371-16682 *378-17028 *389-17608 *419-19192 *448-20749 *484-22515 484-22521 *489-22768 490-22798	
EF.CON	=	000036	G #84-2176	
EF.NEW	=	000035	G #84-2177	
EF.PWR	=	000034	G #84-2178	
EF.RES	=	000037	G #84-2175	
EF.STA	=	000040	G #84-2174	
EIG.FM	=	006142	#85-2238	
ELV.FM	=	006244	#85-2241	
ERRBLK		002160	G #4-96	
ERRMSG		002156	G #4-96	
ERRNBR		002154	G #4-96	
ERRTYP		002152	G #4-96	

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ERR.CH		015212	#100-2974 453-21010 458-21253
EVL	=	000004 G	#84-2187
ESEND	=	002100	#4-13
ESLOAD	=	000035	#4-13 4-65
E2.TEM	=	011156	#88-2399 *94-2693 *328-14370 *333-14673 340-15017 *341-15071 *347-15371 *353-15703 *355-15816 *371-16683 *378-17029 *389-17609 *419-19193 *448-20750 *484-22516 484-22522 *489-22747 *489-22748 490-22799
FIND.C		015430	#102-3081 464-21514 465-21604
FIRST.		016040	#108-3333 241-9948 248-10324 256-10729 297-12792 313-13613 327-14320 327-14342 354-15769
FIV.FM	=	0060C2	#85-2235 193-7503 215-8624 237-9715 237-9732 237-9749 241-9961 242-9980 242-9997 242-10014 243-10035 243-10075 244-10093 244-10122 248-10337 249-10356 249-10373 249-10390 250-10411 250-10451 251-10469 251-10498 257-10746 257-10761 257-10778 257-10795 258-10816 259-10860 259-10874 259-10903 264-11115 264-11130 265-11153 265-11171 265-11189 273-11594 282-12025 283-12079 284-12116 288-12351 288-12368 289-12388 289-12417 290-12463 293-12596 293-12612 294-12632 294-12648 298-12811 298-12830 298-12851 328-14406 348-15421 379-17065 407-18580 412-18840 426-19530 431-19834 434-19995 466-21647 471-21926 478-22220
FMT.1	=	004222	#85-2205 201-7875 497-23167 507-23694 511-23899 516-24140
FMT.10	=	004676	#85-2214 308-13359
FMT.11	=	004750	#85-2215 193-7508
FMT.12	=	005004	#85-2216 298-12816 298-12835 298-12856
FMT.13	=	005034	#85-2217 310-13455
FMT.14	=	005110	#85-2218 366-16415
FMT.15	=	005150	#85-2219 402-18313
FMT.16	=	005216	#85-2220 205-8094 210-8343 216-8635 223-9018 271-11466 272-11530 274-11605
FMT.17	=	005306	#85-2221 187-7243
FMT.18	=	005342	#85-2222
FMT.19	=	005412	#85-2223 454-21031 459-21274 497-23184
FMT.2	=	004266	#85-2206 200-7824 216-8669 233-9544 315-13705 383-17293 408-18595 505-23594
FMT.20	=	005462	#85-2224 454-21038 459-21281
FMT.21	=	005534	#85-2225 464-21531 464-21555 466-21625 466-21653 471-21907 472-21936 477-22197 478-22226
FMT.22	=	005606	#85-2226 485-22566 491-22863
FMT.23	=	005652	#85-2227 180-6861
FMT.24	=	005722	#85-2228 180-6848 180-6875 180-6882 180-6889 180-6896 181-6907 181-6914 181-6921 181-6928 181-6935 181-6942 181-6949 182-6961 182-6970 182-6977
FMT.25	=	005746	#85-2229 226-9175
FMT.26	=	005764	#85-2230 226-9185
FMT.3	=	004354	#85-2207
FMT.4	=	004400	#85-2208 277-11776
FMT.5	=	004430	#85-2209 379-17072 402-18284 402-18320 403-18355 449-20794
FMT.6	=	004532	#85-2210 419-19228
FMT.7	=	004562	#85-2211 284-12164 323-14131 430-19751 430-19771 431-19814
FMT.8	=	004610	#85-2212 520-24321 520-24332
FMT.9	=	004644	#85-2213 289-12422 290-12468 334-14727 413-18910 520-24307
FNC.1	=	007726	#87-2340 233-9530
FNC.10	=	010050	#87-2349 288-12347 288-12363 288-12380 289-12413 290-12459
FNC.11	=	010060	#87-2350 288-12364 293-12592 293-12608 293-12624 294-12644
FNC.12	=	010100	#87-2351 298-12807 298-12826 298-12847
FNC.13	=	010112	#87-2352 353-15739 412-18839 425-19482 425-19504 425-19521
FNC.14	=	010122	#87-2353 328-14401
FNC.15	=	010136	#87-2354 342-15142 364-16286 412-18844
FNC.16	=	010150	#87-2355 347-15413
FNC.17	=	010162	#87-2356 364-16284 366-16407 412-18838 413-18903

SYMBOL	VALUE	REFERENCES
FNC.18	= 010172	#87-2357 390-17657 396-17979
FNC.19	= 010204	#87-2358 437-20111
FNC.2	= 007742	#87-2341 237-9711 237-9728 237-9745 282-12039 283-12075
FNC.21	= 010216	#87-2359 308-13352
FNC.22	= 010230	#87-2360 353-15731
FNC.23	= 010242	#87-2361 190-7361 215-8620 273-11590
FNC.3	= 007750	#87-2342 199-7806 200-7864 264-11111 265-11167 265-11185 266-11210 439-20252 443-20483
FNC.4	= 007756	#87-2343 241-9957 242-1010 243-10031 243-10071 244-10089 244-10118 444-20492
FNC.5	= 007774	#87-2344 204-8076 210-8329 248-10333 249-10386 250-10407 250-10447 251-10465 251-10494
		252-10519 257-10791 258-10812 259-10856 259-10870 259-10899 271-11452 272-11516 372-16734
		373-16760 379-17060 390-17653 402-18271 402-18303 403-18342 443-20485
FNC.6	= 010004	#87-2345 193-7498 204-8075 210-8328 256-10738 260-10924 271-11451 272-11515 308-13347
		383-17276 396-17975 407-18578 443-20484
FNC.7	= 010012	#87-2346 264-11110 265-11166 265-11184 266-11209 498-23214
FNC.8	= 010022	#87-2347 277-11767 277-11770 282-12003 282-12020
FNC.9	= 010036	#87-2348
FORCE.	015602	#104-3183 464-21507 465-21595 471-21883 477-22169
FOR.EC	015760	#106-3277 504-23513 505-23556 506-23619 507-23665 515-24072
FOR.FM	= 006046	#85-2234 277-11771 282-12007 282-12043 302-13031 302-13046 310-13442 315-13691 366-16409
		413-18904 425-19484 425-19506 431-19849 437-20115 440-20287 441-20326 449-20780 464-21549
		471-21901 477-22191 497-23147 498-23217 511-23890
FSAU	= 000015	#4-13
FSAUTO	= 000020	#4-13
FSEGN	= 000040	#4-13 4-39 5-226 5-274 5-305 5-337
FSCLEA	= 000007	#4-13
FSDU	= 000016	#4-13
FSEND	= 000041	#4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13
		4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13 4-13
		5-247 5-288 5-337
FSHARD	= 000004	#4-13 5-226 5-247
FSHW	= 000013	#4-13 4-119 4-139
FSINIT	= 000006	#4-13
FSJMP	= 000050	#4-13
FSMOD	= 000000	#4-13 4-39 5-337
FSMSG	= 000011	#4-13
FSPROT	= 000021	#4-13 5-305 5-311
FSPWR	= 000017	#4-13
FSRPT	= 000012	#4-13
FSSEG	= 000003	#4-13
FSOFT	= 000005	#4-13 5-274 5-288
FSSRV	= 000010	#4-13
FSSUB	= 000002	#4-13
FSSW	= 000014	#4-13 4-150 4-162
FSTEST	= 000001	#4-13
GD.BLK	016062	#109-3382 371-16645 378-16995 382-17240 383-17261 388-17568 395-17920 395-17937 401-18242
		407-18540 407-18557 418-19149 424-19434 424-19456 440-20273 440-20308 441-20335 443-20463
		443-20473 448-20720 448-20725 453-20999 458-21242 463-21480 464-21508 465-21596 470-21835
		471-21884 476-22121 477-22174 482-22437 484-22539 489-22762 490-22836 495-23077 510-23849
		511-23864 515-24075 515-24086
GOOD.B	011564	#79-1881 91-2558 109-3384 *186-7181 *341-15115
GSCNTO	= 000200	#4-13
GSDELTA	= 000372	#4-13

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ISSRV	=	000041	#4-13
ISSUB	=	000041	#4-13
ISTST	=	000041	#4-13
JSJMP	=	000167	#4-13
LAST.B		016114	#110-3431 352-15652 352-15674 354-15787 430-19790
LAU		004210	#43-181 43-198
LAUTO		004164	#43-83 43-100
L(CLEAN		105106	#522-24397 522-24416
LDU		004176	#43-131 43-148
LD.LNG		017124	#121-3963 339-15006 484-22519 490-22796
LINIT		024714	#185-7119 187-7270
LOAD.S		013604	#88-2406 94-2720
LOE	=	040000	G #85-2203
LOT	=	000010	G #84-2188
LRPT		004152	#43-32 43-49
LST.AR		011576	#79-1888 *186-7183 *186-7198 *186-7199 *187-7225 *187-7226 289-12394 289-12435 332-14602
LST.BL		011572	342-15125 342-15134 412-18823 412-18827 504-23540 #79-1886 110-3433 *186-7184 *186-7204 *186-7205 *186-7206 *187-7233 *187-7234 *187-7235 354-15757 429-19715 430-19794 431-19813 434-19972 436-20097 504-23514 505-23563 506-23622 506-23639 507-23662 519-24274
LST.DU		013134	#79-1900 182-6965 *187-7239 *187-7241 193-7477
LSACP		002110	G #4-65
LSAPT		002036	G #4-65
LSAU		004212	G 4-65 #43-198
LSAUT		002070	G #4-65
LSAUTO		004166	G 4-65 #43-100
LSCCP		002106	G #4-65
LSCLEA		105116	G 4-65 #522-24415
LSCO		002032	G #4-65
LSDEPO		002011	G #4-65
LSDESC		002130	G 4-65 #4-87
LSDESP		002076	G #4-65
LSDEVP		002060	G #4-65
LSDISP		002164	G 4-65 #4-103
LSDLY		002116	G #4-65 93-2641 93-2667 93-2680 190-7347 196-7636 200-7841 236-9693 243-10043 250-10419 258-10824 264-11096 301-13010 307-13284 314-13623 314-13647 314-13660 327-14325 327-14346 328-14361 333-14629 333-14652 333-14664 340-15022 340-15045 340-15058 346-15326 346-15346 347-15362 352-15657 352-15678 353-15694 354-15791 355-15807 362-16159 365-16324 370-16623 371-16662 371-16674 377-16973 378-17008 378-17020 389-17589 389-17600 418-19129 418-19165 419-19184 448-20730 448-20741 483-22463 484-22526 490-22804
LSDTP		002040	G #4-65
LSDTYP		002034	G #4-65
LSDU		004200	G 4-65 #43-148
LSDUT		002072	G #4-65
LSDVTY		002122	G 4-65 #4-80
LSEF		002052	G #4-65
LSENVI		002044	G #4-65
LSERRT		002152	G 4-65 #4-96
LSETP		002102	G #4-65
LSEXP1		002046	G #4-65
LSEXP4		002064	G #4-65
LSEXP5		002066	G #4-65

ML11 CREATED BY MACRO ON 22-DEC-80 AT 12:19

PAGE 8
CREF

SEQ 0522

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
LSHARD		002402 G	4-65 5-226 #5-226
LSHIME		002120 G	#4-65
LSHPCP		002016 G	#4-65
LSHPTP		002022 G	#4-65
LSHW		002356 G	4-65 4-119 #4-119
LSICP		002104 G	#4-65
LSINIT		025660 G	4-65 #187-7270
LSLADP		002026 G	#4-65
LSLAST	=	105132 G	4-65 #523-24444
LSLOAD		002100 G	#4-65
LSLUN		002074 G	#4-65
LSMREV		002050 G	#4-65
LSNAME		002000 G	#4-65
LSPRIO		002042 G	#4-65
LSPROT		002750 G	4-65 #5-305
LSPRT		002112 G	#4-65
LSREPP		002062 G	#4-65
LSREV		002010 G	#4-65
LSRPT		004154 G	4-65 #43-49
LSOFT		002720 G	4-65 5-274 #5-274
LSSPC		002056 G	#4-65
LSSPCP		002020 G	#4-65
LSSPTP		002024 G	#4-65
LSSTA		002030 G	#4-65
LSSW		002376 G	4-65 4-150 #4-150
LSTEST		002114 G	#4-65
LSIML		002014 G	#4-65
LSUNIT		002012 G	#4-65 185-7113 185-7129 186-7165
L1000		002374	4-119 #4-139
L10001		002400	4-150 #4-162
L10002		002476	5-226 #5-247
L10003		002726	5-274 #5-288
MEM.AR	=	010630	#88-2390
ML.DUT		013602	#84-2145 93-2658 94-2729 157-5790 158-5815 161-5978 162-6003 166-6189 167-6217 *186-7176 187-7248 189-7335 193-7487 196-7629 199-7790 199-7795 199-7797 200-7817 200-7823 200-7832 200-7851 200-7856 201-7874 203-8027 209-8280 214-8554 214-8572 220-8867 229-9324 230-9349 230-9369 233-9509 236-9685 241-9943 243-10055 244-10102 248-10319 250-10431 251-10478 256-10724 258-10836 259-10883 263-11083 270-11425 272-11493 273-11561 277-11749 281-11973 281-11980 281-11987 282-12051 283-12062 283-12088 283-12095 284-12130 284-12134 284-12141 288-12334 289-12400 290-12446 293-12580 297-12787 301-12992 306-13257 313-13607 314-13638 320-13963 327-14312 327-14337 332-14610 333-14641 339-14981 340-15034 346-15312 346-15338 352-15642 352-15669 354-15764 354-15782 362-16145 365-16311 370-16610 371-16639 371-16651 377-16961 378-16989 378-17000 382-17234 382-17246 388-17564 388-17573 394-17868 395-17933 395-17942 401-18227 406-18521 407-18553 411-18791 412-18857 413-18876 417-19102 418-19141 418-19154 424-19428 424-19441 429-19722 430-19785 434-19976 436-20091 439-20261 440-20303 443-20457 443-20468 447-20686 453-20995 457-21213 463-21489 465-21581 470-21866 476-22152 482-22433 482-22442 484-22535 489-22751 490-22832 495-23043 498-23203 504-23505 504-23544 506-23611 506-23651 507-23674 510-23839 511-23860 515-24064 515-24082 519-24279
ML.LUN		013600	#84-2144 *185-7127 *185-7128 185-7129 185-7132 *185-7159 *185-7160 186-7165 186-7168 187-7242 190-7367 194-7526 196-7661 201-7887 205-8113 210-8366 216-8676 223-9041 234-9559 237-9760 244-10134 252-10532 260-10937 266-11222 274-11625 278-11795 310-13467

REFERENCES

284-12133	288-12333	288-12336	288-12337	288-12338	288-12339	288-12340	288-12356	288-12373
289-12399	289-12402	289-12403	289-12404	289-12405	289-12406	290-12445	290-12448	290-12449
290-12450	290-12451	290-12452	293-12579	293-12582	293-12583	293-12584	293-12585	293-12601
293-12617	294-12637	297-12786	297-12789	297-12790	297-12793	297-12794	297-12796	298-12840
301-12991	301-12994	301-12995	301-12996	301-13003	301-13005	301-13007	301-13017	302-13036
306-13256	306-13259	306-13260	307-13266	307-13281	307-13291	313-13606	313-13609	313-13610
313-13612	313-13614	313-13615	314-13632	314-13633	314-13637	314-13640	314-13641	314-13642
314-13644	314-13654	314-13657	314-13667	320-13962	320-13965	320-13966	321-14016	321-14018
321-14019	321-14020	321-14021	327-14311	327-14314	327-14315	327-14316	327-14317	327-14318
327-14319	327-14321	327-14332	327-14336	327-14339	327-14340	327-14341	327-14343	328-14358
328-14368	328-14369	328-14370	332-14609	332-14612	332-14613	332-14614	332-14615	332-14616
332-14617	332-14618	333-14623	333-14624	333-14625	333-14636	333-14640	333-14643	333-14644
333-14645	333-14646	333-14647	333-14648	333-14649	333-14660	333-14661	333-14671	333-14672
333-14673	339-14980	339-14983	339-14984	339-14985	339-14986	339-14987	339-14988	340-15015
340-15016	340-15017	340-15018	340-15029	340-15033	340-15036	340-15037	340-15038	340-15039
340-15040	340-15041	340-15042	340-15054	340-15055	341-15069	341-15070	341-15071	346-15311
346-15314	346-15315	346-15318	346-15319	346-15320	346-15322	346-15333	346-15337	346-15340
346-15341	346-15343	346-15354	346-15355	347-15369	347-15370	347-15371	352-15641	352-15644
352-15645	352-15648	352-15649	352-15650	352-15651	352-15653	352-15664	352-15668	352-15671
352-15672	352-15673	352-15675	353-15690	353-15691	353-15701	353-15702	353-15703	354-15759
354-15760	354-15761	354-15763	354-15766	354-15767	354-15768	354-15770	354-15774	354-15781
354-15784	354-15785	354-15786	354-15788	355-15803	355-15804	355-15814	355-15815	355-15816
362-16144	362-16147	362-16148	362-16151	362-16152	362-16153	362-16155	362-16156	362-16166
363-16201	363-16209	363-16210	363-16215	363-16224	363-16225	363-16229	363-16237	363-16238
363-16243	364-16256	364-16257	364-16262	364-16263	364-16270	365-16310	365-16313	365-16314
365-16316	365-16317	365-16318	365-16320	365-16321	365-16331	365-16347	365-16360	366-16366
366-16368	366-16370	366-16372	366-16374	366-16380	366-16382	366-16388	366-16396	370-16609
370-16612	370-16613	370-16615	370-16616	370-16617	370-16619	370-16630	371-16638	371-16641
371-16642	371-16644	371-16646	371-16647	371-16650	371-16653	371-16654	371-16659	371-16670
371-16671	371-16681	371-16682	371-16683	377-16960	377-16963	377-16964	377-16965	377-16966
377-16967	377-16969	378-16984	378-16988	378-16991	378-16992	378-16993	378-16996	378-16997
378-16999	378-17002	378-17003	378-17005	378-17016	378-17017	378-17027	378-17028	378-17029
382-17233	382-17236	382-17237	382-17238	382-17241	382-17242	382-17245	383-17252	383-17253
383-17262	383-17263	388-17563	388-17566	388-17567	388-17569	388-17570	388-17572	388-17575
389-17580	389-17586	389-17596	389-17597	389-17607	389-17608	389-17609	394-17867	394-17870
394-17871	395-17921	395-17922	395-17932	395-17935	395-17936	395-17938	395-17939	395-17941
395-17944	395-17945	401-18226	401-18229	401-18230	401-18243	401-18244	406-18520	406-18523
406-18524	407-18541	407-18542	407-18552	407-18555	407-18556	407-18558	407-18559	411-18790
411-18793	411-18794	411-18795	411-18796	411-18798	411-18799	411-18800	411-18801	411-18802
411-18803	411-18805	411-18807	411-18808	411-18809	411-18810	412-18815	412-18816	412-18818
412-18856	412-18859	412-18860	412-18861	412-18862	412-18864	412-18865	412-18866	413-18871
413-18872	413-18873	413-18875	413-18878	413-18879	413-18880	413-18882	413-18885	413-18886
413-18889	413-18890	413-18891	413-18892	413-18894	413-18898	417-19101	417-19104	417-19105
417-19117	417-19118	417-19119	417-19121	418-19136	418-19140	418-19143	418-19144	418-19148
418-19150	418-19151	418-19153	418-19156	418-19157	418-19162	418-19176	418-19177	419-19191
419-19192	419-19193	424-19427	424-19430	424-19431	424-19432	424-19435	424-19436	424-19440
424-19443	424-19444	424-19454	424-19455	424-19457	424-19462	424-19463	425-19515	429-19721
429-19724	429-19725	429-19726	429-19727	429-19728	429-19729	429-19730	429-19731	429-19733
430-19757	430-19784	430-19787	430-19788	430-19789	430-19791	430-19792	430-19796	431-19820
431-19822	431-19823	434-19975	434-19978	434-19979	434-19980	434-19981	434-19982	434-19983
434-19984	436-20090	436-20093	436-20094	436-20095	436-20096	436-20097	437-20102	437-20103
437-20105	439-20260	439-20263	440-20268	440-20269	440-20274	440-20275	440-20277	440-20293

REFERENCES

440-20302	440-20305	440-20306	440-20307	440-20309	440-20310	440-20312	441-20336	441-20337
443-20456	443-20459	443-20460	443-20461	443-20464	443-20465	443-20467	443-20470	443-20471
443-20472	443-20474	443-20475	443-20477	447-20685	447-20688	447-20689	447-20690	447-20701
448-20706	448-20709	448-20710	448-20713	448-20714	448-20717	448-20718	448-20721	448-20722
448-20724	448-20726	448-20737	448-20738	448-20748	448-20749	448-20750	453-20994	453-20997
453-20998	453-21000	453-21001	457-21212	458-21219	458-21220	458-21243	458-21244	463-21478
463-21481	463-21482	463-21488	463-21491	463-21492	464-21509	464-21510	464-21511	465-21580
465-21583	465-21584	465-21597	465-21598	465-21599	470-21833	470-21836	470-21837	470-21865
470-21868	470-21869	471-21885	471-218 6	471-21887	471-218 9	471-21891	476-22119	476-22122
476-22123	476-22151	476-22154	476-22155	477-22170	477-22172	477-22173	477-22175	477-22176
477-22177	477-22179	477-22181	482-22432	482-22435	48 22436	482-22438	482-22439	482-22441
482-22444	482-22445	482-22448	4 2-22449	483-22454	483-224 0	483-22475	483-22490	484-22520
484-22521	484-22522	484-22523	484-22534	484-22537	484-225 3	484-22540	484-22541	484-22542
489-22750	489-22753	489-22754	489-227 3	489-227 4	490-22797	490-22798	490-22799	490-22801
490-22816	490-22827	490-2 931	490-2 4	490-2 35	490-2 37	490-2 38	490-22939	495-23042
495-23045	495-23046	495-23061	495-2 2	495-2 67	495-2 8	495-23071	495-23072	495-23075
495-23076	495-23078	495-23079	495-23 0	496-23 8	496-23095	496-23102	496-23109	496-23116
496-23127	497-23165	498-23202	498-23 05	498-23206	498-23207	498-23222	504-23504	504-23507
504-23508	504-23514	504-23515	504-23516	504-23517	504-23518	504-23520	504-23543	505-23550
505-23551	505-23564	505-23567	505-23569	505-23570	505-23571	505-23572	505-23574	506-23610
506-23613	506-23614	506-23620	506-23621	506-23622	506-23623	506-23624	506-23628	506-23644
506-23650	506-23653	506-23654	507-23662	507-23666	507-23667	507-23668	507-23669	507-23673
507-23676	507-23677	507-23678	510-23838	510-23841	510-23842	510-23843	510-23844	510-23845
510-23846	510-23847	510-23850	510-23851	510-23855	511-23862	511-23863	511-23867	511-23870
511-23872	511-23873	511-23877	511-23895	515-24063	515-24066	515-24067	515-24073	515-24 76
515-24077	515-24081	515-24084	515-24085	515-24093	515-24094	515-24096	515-24097	515-24100
515-24101	515-24103	515-24104	515-24108	516-24136	519-24278	519-24281	519-24282	519-24283
519-24284	519-24285	519-24286	519-24287	519-24288	519-24290	522-24397		

MSGH1 002476
MSGH2 002512
MSGH3 002540
MSGH4 002563
MSGH5 002614
MSGH6 002646
MSGH7 002672
MSGS1 002726
MSDUMP 024062
NIB.SA 011152

5-239	#5-256							
5-240	#5-257							
5-241	#5-258							
5-242	#5-259							
5-243	#5-260							
5-244	#5-261							
5-245	#5-262							
5-285	#5-297							
179-6800	#180-6855							
#79-1872	88-2397	88-2398	88-2399	89-2433	89-2439	89-2445	89-2451	89-2457
89-2463	90-2477	90-2483	90-2493	90-2502	113-3588	113-3595	113-3606	114-3618
114-3630	114-3637	114-3648	114-3656	114-3668	115-3680	118-3817	118-3820	118-3828
118-3830	118-3832	118-3839	118-3845	118-3847	118-3853	118-3857	*121-3980	*121-3987
*121-3991	*122-4003	*122-4004	*122-4007	*122-4014	*122-4018	*122-4026	*122-4027	*122-4031
*122-4039	*122-4040	449-20785						

NIN.FM = 006166
ONEFIL = 000001
ONE.FM = 006016
OP.NUM 011560

#85-2239								
#2-4	2-1307	3-1308	4-34	4-164	5-190	5-353		
#85-2231	180-6857	196-7650						
#79-1878	*186-7180	186-7194	186-7200	186-7215	187-7227	233-9514	233-9542	289-12433
362-16141	364-16302							

OSAPTS = 000001
OSAU = 000001
OSBGNR = 000001
OSBGNS = 000001

#4-13	#4-46	4-65						
#4-13	#4-46	4-65						
#4-13	#4-46	4-65						
#4-13	#4-46	4-65						

ML11 CREATED BY MACRO ON 22-DEC-80 AT 12:19

PAGE 12
CREF

SEQ 0526

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
OSDU	=	000001	#4-13 #4-46 4-65
OSERRT	=	000001	#4-13 #4-46 4-65
OSGNSW	=	000001	#4-13 #4-46 4-65
OSPOIN	=	000001	#4-13 #4-46 4-46 4-65
OSSETU	=	000001	#4-13 #4-46 4-65
PAR.DI		011566	#79-1883 *186-7174 229-9319 439-20248
PD.TEM		013120	#79-1892 *93-2676 94-2701 *333-14660 334-14683 *340-15054 341-15078 *346-15354 347-15377
			*353-15690 353-15709 *355-15803 355-15822 *362-16166 362-16173 *365-16331 365-16337 *371-16670
			372-16693 *378-17016 379-17039 *389-17596 389-17619 *418-19176 419-19199 447-20682 *448-20737
			483-22474 *483-22475 490-22815 *490-22816
PHR.1	=	007366	#87-2325 241-9959 242-9978 244-10091 248-10335 249-10354 251-10467 257-10744 257-10759
			259-10872 265-11151 277-11769 282-12025 282-12022 289-12386 290-12461 294-12646 298-12828
			302-13029 302-13044 322-14082 323-14103 323-14120 426-19528 431-19847 434-19993 437-20113
			464-21522 465-21612
PHR.10	=	007564	#87-2334 342-15143 364-16287 412-18845
PHR.11	=	007576	#87-2335 497-23158 504-23526
PHR.12	=	007614	#87-2336 484-22555
PHR.13	=	007636	#87-2337 491-22856
PHR.14	=	007662	#87-2338 180-6856
PHR.15	=	007714	#87-2339 226-9174 226-9184
PHR.2	=	007404	#87-2326 237-9713 242-9995 243-10073 244-10120 249-10371 250-10449 251-10496 257-10776
			259-10858 259-10901 282-12041 283-12077 284-12114 288-12349 293-12594 298-12809 431-19832
PHR.3	=	007422	#87-2327 196-7649
PHR.4	=	007454	#87-2328 190-7360 199-7808 204-8078 210-8331 215-8623 216-8661 271-11454 272-11518
			273-11593 315-13687 328-14404 348-15419 353-15733 372-16738 373-16764 402-18273 402-18305
			403-18344 407-18575 505-23581 519-24296
PHR.5	=	007472	#87-2329 193-7501 237-9730 237-9747 242-9992 242-10012 243-10033 249-10368 249-10388
			250-10409 252-10521 257-10773 257-10793 258-10814 260-10926 264-11113 264-11125 264-11128
			265-11169 265-11187 288-12366 289-12415 293-12610 293-12626 298-12849 322-14044 322-14061
			429-19740 464-21545 516-24129
PHR.6	=	007504	#87-2330 242-9975 249-10351 257-10756 264-11144 284-12157
PHR.7	=	007516	#87-2331 284-12153
PHR.8	=	007540	#87-2332 425-19483
PHR.9	=	007552	#87-2333 425-19505
PNT	=	001000	G #84-2194
PRI	=	002000	G #85-2199
PRI00	=	000000	G #84-2186
PRI01	=	000040	G #84-2185
PRI02	=	000100	G #84-2184
PRI03	=	000140	G #84-2183
PRI04	=	000200	G #84-2182
PRI05	=	000240	G #84-2181
PRI06	=	000300	G #84-2180
PRI07	=	000340	G #84-2179
PRSN		002376	G #4-160 225-9134 226-9145
PTBL.P		011556	#79-1876 *185-7134 185-7135 185-7137 185-7138 185-7140 *186-7170 186-7171 186-7173
			186-7175 186-7177 186-7185 187-7236
P.AAA		004222	#62-912 85-2205
P.AAB		004266	#62-924 85-2206
P.AAC		004354	#62-942 85-2207
P.AAD		004400	#62-949 85-2208
P.AAE		004430	#62-957 85-2209

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.AAF		004532	#63-983 85-2210
P.AAG		004562	#63-991 85-2211
P.AAH		004610	#63-999 85-2212
P.AAI		004644	#63-1009 85-2213
P.AAJ		004676	#63-1018 85-2214
P.AAK		004750	#64-1036 85-2215
P.AAL		005004	#64-1046 85-2216
P.AAM		005034	#64-1054 85-2217
P.AAN		005110	#64-1069 85-2218
P.AAO		005150	#65-1084 85-2219
P.AAP		005216	#65-1097 85-2220
P.AAQ		005306	#65-1116 85-2221
P.AAR		005342	#65-1126 85-2222
P.AAS		005412	#66-1144 85-2223
P.AAT		005462	#66-1158 85-2224
P.AAU		005534	#66-1172 85-2225
P.AAV		005606	#66-1186 85-2226
P.AAW		005652	#67-1202 85-2227
P.AAX		005722	#67-1216 85-2228
P.AAY		005746	#67-1223 85-2229
P.AAZ		005764	#67-1228 85-2230
P.ABA		006016	#67-1237 85-2231
P.ABB		006024	#67-1239 85-2232
P.ABC		006034	#67-1242 85-2233
P.ABD		006046	#68-1250 85-2234
P.ABE		006062	#68-1254 85-2235
P.ABF		006100	#68-1259 85-2236
P.ABG		006120	#68-1265 85-2237
P.ABH		006142	#68-1271 85-2238
P.ABI		006166	#68-1278 85-2239
P.ABJ		006214	#68-1286 85-2240
P.ABK		006244	#68-1294 85-2241
P.ABL		006276	#69-1307 85-2242
P.ABM		006302	#69-1309 85-2243
P.ABN		006314	#69-1313 85-2244
P.ABO		006322	#69-1315 85-2245
P.ABP		006330	#69-1317 85-2246
P.ABQ		006336	#69-1319 85-2247
P.ABR		006344	#69-1321 85-2248
P.ABS		006360	#69-1325 85-2249
P.ABT		006374	#69-1329 85-2250
P.ABU		006406	#69-1333 86-2255
P.ABV		006416	#69-1336 86-2256
P.ABW		006426	#69-1339 86-2257
P.ABX		006436	#69-1342 86-2258
P.ABY		006442	#69-1344 86-2259
P.ABZ		006454	#69-1348 86-2260
P.ACA		006462	#69-1350 86-2261
P.ACB		006470	#69-1352 86-2262
P.ACC		006502	#70-1360 86-2263
P.ACD		006506	#70-1362 86-2264
P.ACE		006514	#70-1364 86-2265

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.ACF		006522	#70-1366 86-2266
P.ACG		006530	#70-1368 86-2267
P.ACH		006544	#70-1372 86-2268
P.ACI		006552	#70-1374 86-2269
P.ACJ		006560	#70-1376 86-2270
P.ACK		006574	#70-1380 86-2271
P.ACL		006602	#70-1382 86-2272
P.ACM		006610	#70-1384 86-2273
P.ACN		006624	#70-1388 86-2274
P.ACO		006632	#70-1390 86-2275
P.ACP		006646	#70-1394 86-2276
P.ACQ		006654	#70-1396 86-2277
P.ACR		006660	#70-1398 86-2278
P.ACS		006666	#70-1400 86-2279
P.ACT		006674	#70-1402 86-2280
P.ACU		006700	#70-1404 86-2281
P.ACV		006710	#70-1407 86-2282
P.ACW		006716	#70-1409 86-2283
P.ACX		006726	#71-1416 86-2284
P.ACY		006732	#71-1418 86-2285
P.ACZ		006746	#71-1422 86-2286
P.ADA		006754	#71-1424 86-2287
P.ADB		006764	#71-1427 86-2288
P.ADC		006772	#71-1429 86-2289
P.ADD		007004	#71-1433 86-2290
P.ADE		007016	#71-1437 86-2291
P.ADF		007026	#71-1440 86-2292
P.ADG		007036	#71-1443 86-2293
P.ADH		007046	#71-1446 86-2294
P.ADI		007054	#71-1448 86-2295
P.ADJ		007066	#71-1452 86-2296
P.ADK		007074	#71-1454 86-2297
P.ADL		007104	#71-1457 86-2298
P.ADM		007114	#71-1460 86-2299
P.ADN		007122	#71-1462 86-2300
P.ADO		007130	#71-1464 86-2301
P.ADP		007140	#71-1467 86-2302
P.ADQ		007152	#72-1475 86-2303
P.ADR		007156	#72-1477 86-2304
P.ADS		007174	#72-1482 86-2305
P.ADT		007210	#72-1486 86-2306
P.ADU		007216	#72-1488 87-2311
P.ADV		007224	#72-1490 87-2312
P.ADW		007232	#72-1492 87-2313
P.ADX		007240	#72-1494 87-2314
P.ADY		007246	#72-1496 87-2315
P.ADZ		007252	#72-1498 87-2316
P.AEA		007260	#72-1500 87-2317
P.AEB		007272	#72-1504 87-2318
P.AEC		007302	#72-1507 87-2319
P.AED		007310	#72-1509 87-2320
P.AEE		007316	#72-1511 87-2321

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
P.AEF	007330	#72-1515 87-2322
P.AEG	007340	#72-1518 87-2323
P.AEH	007346	#72-1520 87-2324
P.AEI	007366	#73-1530 87-2325
P.AEJ	007404	#73-1535 87-2326
P.AEK	007422	#73-1540 87-2327
P.AEL	007454	#73-1549 87-2328
P.AEM	007472	#73-1554 87-2329
P.AEN	007504	#73-1558 87-2330
P.AEO	007516	#73-1562 87-2331
P.AEP	007540	#73-1568 87-2332
P.AEQ	007552	#73-1572 87-2333
P.AER	007564	#73-1576 87-2334
P.AES	007576	#74-1584 87-2335
P.AET	007614	#74-1589 87-2336
P.AEU	007636	#74-1595 87-2337
P.AEV	007662	#74-1602 87-2338
P.AEW	007714	#74-1611 87-2339
P.AEX	007726	#74-1615 87-2340
P.AEY	007742	#74-1619 87-2341
P.AEZ	007750	#74-1621 87-2342
P.AFA	007756	#74-1623 87-2343
P.AFB	007774	#74-1628 87-2344
P.AFC	010004	#74-1631 87-2345
P.AFD	010012	#74-1633 87-2346
P.AFE	010022	#75-1640 87-2347
P.AFF	010036	#75-1644 87-2348
P.AFG	010050	#75-1648 87-2349
P.AFH	010060	#75-1651 87-2350
P.AFI	010100	#75-1657 87-2351
P.AFJ	010112	#75-1661 87-2352
P.AFK	010122	#75-1664 87-2353
P.AFL	010136	#75-1668 87-2354
P.AFM	010150	#75-1672 87-2355
P.AFN	010162	#75-1676 87-2356
P.AFO	010172	#75-1679 87-2357
P.AFP	010204	#75-1683 87-2358
P.AFQ	010216	#75-1687 87-2359
P.AFR	010230	#75-1691 87-2360
P.AFS	010242	#76-1699 87-2361
P.AFT	010256	#76-1703 87-2362
P.AFU	010266	#76-1706 88-2367
P.AFV	010274	#76-1708 88-2368
P.AFW	010302	#76-1710 88-2369
P.AFX	010310	#76-1712 88-2370
P.AFY	010316	#76-1714 88-2371
P.AFZ	010324	#76-1716 88-2372
P.AGA	010332	#76-1718 88-2373
P.AGB	010340	#76-1720 88-2374
P.AGC	010346	#76-1722 88-2375
P.AGD	010354	#76-1724 88-2376
P.AGE	010362	#76-1726 88-2377

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.AGF		010370	#76-1728 88-2378
P.AGG		010376	#76-1730 88-2379
P.AGH		010404	#76-1732 88-2380
P.AGI		010412	#76-1734 88-2381
P.AGJ		010420	#76-1736 88-2382
P.AGK		010430	#76-1739 88-2383
P.AGL		010436	#76-1741 88-2384
P.AGM		010444	#76-1743 88-2385
P.AGN		010454	#76-1746 88-2386
P.AGO		010464	#77-1753 88-2387
P.AGP		010526	#77-1765 88-2388
P.AGD		010570	#77-1777 88-2389
P.AGR		010630	#77-1788 88-2390
P.AGS		010672	#77-1800 88-2391
P.AGT		010732	#78-1815 88-2392
P.AGU		010772	#78-1826 88-2393
P.AGV		011026	#78-1836 88-2394
P.AGW		011060	#78-1845 88-2395
P.AGX		011106	#78-1853 88-2396
P.CAL		013144	#79-1906 98-2885 *98-2886 *98-2887 101-3013 *453-21008 454-21035 *458-21251 459-21278
P.GEN		013152	#79-1909 101-2985 101-3013 *453-21005 454-21028 *458-21248 458-21267
RAS.IN		013124	#79-1895 *186-7190 *186-7210 341-15117 411-18787 411-18797
RD.CS1		017532	#125-4179 177-6706 222-8945
RD.DA		020066	#131-4494 177-6716 222-8959
RD.DAT		013130	#79-1897 *125-4194 125-4195 *128-4352 128-4353 *131-4509 131-4510 *134-4667 134-4668
			*137-4826 137-4827 *140-4986 140-4987 *144-5170 144-5171 *148-5356 148-5357 *151-5492
			151-5493 *154-5625 154-5626 *158-5810 158-5811 *162-5998 162-5999 *167-6211 167-6212
			*170-6369 170-6370 204-8083 210-8336 215-8628 223-9011 271-11459 272-11523 274-11602
RD.DS		022766	#170-6354 178-6760
RD.D1		021654	#157-5781 178-6765
RD.D2		022166	#161-5969 178-6770
RD.D3		022514	#166-6179 178-6775
RD.EE		021530	#154-5623 177-6746
RD.EL		021466	#151-5490 177-6751
RD.ER		017710	#128-4337 177-6711 222-8952
RD.E1		020644	#140-4970 177-6726 222-8973
RD.E2		021064	#144-5153 177-6731 222-8980
RD.MR		020244	#134-4652 177-6721
RD.PA		020436	#137-4810 177-6736 222-8966
RD.PD		021326	#148-5339 177-6741
RD.REG		023462	#176-6679 204-8044 209-8293 215-8584 271-11442 272-11506 273-11582
REG.IN		013136	#79-1902 157-5787 161-5975 166-6186 *273-11555 *273-11577 *274-11622
REG.1	=	010256	#87-2362 180-6867
REG.10	=	010346	#88-2375
REG.11	=	010354	#88-2376
REG.12	=	010362	#88-2377
REG.13	=	010370	#88-2378
REG.14	=	010376	#88-2379 181-6948 497-23160 498-23216
REG.15	=	010404	#88-2380 182-6960 504-23528 505-23582 507-23687
REG.16	=	010412	#88-2381
REG.17	=	010420	#88-2382 180-6895 190-7362
REG.18	=	010430	#88-2383 180-6874

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

CREATED BY MACRO ON 22-DEC-80 AT 12:19

PAGE 18
CREF

M 9

SEQ 0532

REFERENCES

		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103	4-103
		5-239	5-239	5-239	5-239	5-240	5-240	5-240	5-240	5-240
		5-241	5-241	5-241	5-241	5-241	5-242	5-242	5-242	5-242
		5-242	5-243	5-243	5-243	5-244	5-244	5-244	5-244	5-244
		5-245	5-245	5-245	5-247	5-274	5-285	5-285	5-285	5-288
SVCSUB	= 177777	#4-13	#4-21							
SVCTAG	= 177777	#4-13	#4-23	4-139	4-139	4-139	4-162	4-162	4-162	5-247
		5-247	5-247	5-288	5-288	5-288				
SVCTST	= 177777	#4-13	#4-20							
SYNC	= 010526	#88-2388	204-8062	209-8311	215-8608	243-10068	244-10115	250-10444	251-10491	258-10849
		259-10896	302-13040	372-16731	379-17058	383-17273	401-18264	402-18300	403-18339	407-18573
		425-19478	425-19500	425-19519	437-20109	440-20281	440-20316	449-20774	454-21019	458-21258
		464-21520	464-21543	465-21610	466-21640	471-21895	471-21919	477-22185	477-22209	484-22551
		491-22852	497-23156	498-23211	504-23524	507-23683	511-23884	516-24121		
SLSYM	= 010000	#4-13	#4-139	#4-162	#5-247	#5-288				
TEN.FM	= 006214	#85-2240								
THR.FM	= 006034	#85-2233	190-7363	200-7865	223-9007	230-9343	230-9363	230-9383	334-14722	342-15145
		353-15734	355-15847	412-18847	419-19223	430-19746	430-19766	431-19809	439-20253	454-21024
		458-21263	464-21525	466-21619	484-22556	491-22857	497-23161	504-23529	507-23688	516-24131
TIME.O	= 011106	#88-2396								
TRELE.	= 011026	#88-2394	443-20481							
TST.LN	016146	#113-3569	328-14376	334-14697	341-15092	347-15391	353-15723	355-15836	390-17645	419-19213
		449-20769								
TWO.FM	= 006024	#85-2232	216-8663	231-9399	233-9531	353-15740	505-23583	519-24298		
TSARGC	= 000003	#4-65	4-65	#4-65	4-65	4-65	#4-65	4-65	4-65	#4-65
		4-65	4-65	#4-65	4-65	4-65				
TSCODE	= 000130	#5-239	5-239	#5-239	5-239	#5-239	5-239	#5-240	5-240	#5-240
		5-240	#5-240	5-240	#5-241	5-241	#5-241	5-241	#5-241	5-241
		#5-242	5-242	#5-242	5-242	#5-242	5-242	#5-243	5-243	#5-243
		5-243	#5-243	5-243	#5-244	5-244	#5-244	5-244	#5-244	5-244
		#5-245	5-245	#5-245	5-245	#5-245	5-245	#5-285	5-285	#5-285
		5-285	#5-285	5-285						
TSERNR	= 000000	#4-13								
TSEXCP	= 000000	#5-239	5-239	#5-240	5-240	#5-241	5-241	#5-242	5-242	#5-244
		5-244								
TSFREE	105132	523-24439	523-24440	#523-24441						
TSGMAN	= 000000	#4-13								
TSHILI	= 000007	#5-239	5-239	#5-240	5-240	#5-241	5-241	#5-242	5-242	#5-244
		5-244								
TSLAST	= 000000	#4-13								
TSLOLI	= 000000	#5-239	5-239	#5-240	5-240	#5-241	5-241	#5-242	5-242	#5-244
		5-244								
TSLSYM	= 010000	#4-13	4-13	4-139	4-162	5-247	5-288			
TSNEST	= 177777	#4-13	4-39	#4-39	4-39	4-119	#4-119	4-119	4-139	4-139
		4-139	#4-139	4-150	#4-150	4-150	4-162	4-162	4-162	#4-162
		5-226	#5-226	5-226	5-247	5-247	5-247	#5-247	5-274	#5-274

G

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
T13		035042 G	4-103 #245-10159
T14		036310 G	4-103 #252-10553
T15		037556 G	4-103 #260-10958
T16		040456 G	4-103 #266-11243
T17		041652 G	4-103 #274-11646
T18		042110 G	4-103 #278-11815
T19		043260 G	4-103 #285-12199
T2		026322 G	4-103 #194-7546
T20		044220 G	4-103 #291-12500
T21		044614 G	4-103 #294-12670
T22		045262 G	4-103 #299-12892
T23		045620 G	4-103 #303-13079
T24		046774 G	4-103 #311-13492
T25		047552 G	4-103 #316-13741
T26		050676 G	4-103 #324-14171
T27		051454 G	4-103 #329-14436
T28		052440 G	4-103 #335-14768
T29		053552 G	4-103 #342-15171
T3		026530 G	4-103 #197-7684
T30		054346 G	4-103 #348-15447
T31		055742 G	4-103 #356-15896
T32		057500 G	4-103 #367-16449
T33		060512 G	4-103 #373-16788
T34		061446 G	4-103 #380-17121
T35		062100 G	4-103 #384-17337
T36		063136 G	4-103 #391-17715
T37		064044 G	4-103 #397-18028
T38		065050 G	4-103 #404-18406
T39		065476 G	4-103 #408-18630
T4		027222 G	4-103 #201-7908
T40		066464 G	4-103 #414-18939
T41		067516 G	4-103 #420-19279
T42		070270 G	4-103 #426-19559
T43		071224 G	4-103 #432-19884
T44		071434 G	4-103 #435-20027
T45		071620 G	4-103 #437-20137
T46		072316 G	4-103 #441-20360
T47		072610 G	4-103 #444-20519
T48		073502 G	4-103 #450-20831
T49		074206 G	4-103 #455-21086
T5		027644 G	4-103 #205-8134
T50		074624 G	4-103 #459-21315
T51		075676 G	4-103 #467-21689
T52		076450 G	4-103 #472-21972
T53		077250 G	4-103 #478-22263
T54		100272 G	4-103 #485-22604
T55		101142 G	4-103 #491-22897
T56		102244 G	4-103 #498-23246
T57		103466 G	4-103 #508-23729
T58		104072 G	4-103 #512-23931
T59		104540 G	4-103 #517-24173
T6		030302 G	4-103 #211-8391

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
T60		105072 G	4-103 #521-24364
T7		031132 G	4-103 #217-8701
T8		032116 G	4-103 #224-9066
T9		032352 G	4-103 #227-9212
UAM	=	000200 G	#84-2192
WRD.1	=	006276	#85-2242 237-9714 241-9960 242-9976 242-9993 243-10074 244-10121 248-10336 249-10352
			249-10369 250-10450 251-10497 257-10745 257-10757 257-10774 259-10859 259-10902 264-11114
			264-11126 265-11149 288-12350 293-12595 298-12810
WRD.10	=	006406	#86-2255 223-9004 308-13351 334-14719 355-15844 366-16405 379-17062 383-17279 390-17655
			396-17977 419-19220 425-19480 425-19502 434-19990 444-20491 449-20776 471-21899 477-22189
			497-23145 507-23685 511-23888 516-24128
WRD.11	=	006416	#86-2256 237-9712 237-9729 237-9746 243-10072 244-10090 244-10119 250-10448 251-10466
			251-10495 259-10857 259-10871 259-10900 264-11112 282-12004 282-12021 282-12040 283-12076
			283-12109 284-12156 288-12348 288-12365 293-12593 293-12609 293-12625 294-12645 298-12808
			298-12827 302-13028 302-13043 426-19527 431-19831 431-19846 434-19992 437-20112
WRD.12	=	006426	#86-2257 193-7500 199-7807 204-8077 210-8330 215-8622 241-9958 242-10011 243-10032
			248-10334 249-10387 250-10408 252-10520 256-10739 257-10792 258-10813 260-10925 265-11168
			265-11186 271-11453 272-11517 273-11592 277-11768 288-12381 289-12414 290-12460 298-12848
			308-13350 328-14403 348-15418 364-16285 372-16736 373-16762 379-17061 383-17279 390-17654
			396-17976 402-18272 402-18304 403-18343 444-20490 466-21643 471-21922 477-22212
WRD.13	=	006436	#86-2258 199-7804 266-11206 353-15732
WRD.14	=	006442	#86-2259 200-7862 233-9529 266-11207 342-15144 364-16288 412-18846 413-18900 430-19763
			431-19806 496-23139 498-23213 511-23886
WRD.15	=	006454	#86-2260 282-12006 282-12042 283-12078 284-12115 284-12158
WRD.16	=	006462	#86-2261 282-12023 289-12387
WRD.17	=	006470	#86-2262 283-12108 284-12155 431-19830
WRD.18	=	006502	#86-2263 294-12647
WRD.19	=	006506	#86-2264 237-9710 237-9727 237-9744 241-9956 242-10009 243-10030 243-10070 244-10088
			244-10117 248-10332 249-10385 250-10406 250-10446 251-10464 251-10493 252-10518 256-10737
			257-10790 258-10811 258-10851 259-10869 259-10898 260-10923 264-11109 265-11165 265-11183
			266-11208 282-12038 283-12074 288-12346 288-12362 288-12379 289-12412 290-12458 293-12591
			293-12607 293-12623 294-12643 298-12806 298-12825 298-12846 315-13688 372-16733 373-16759
			383-17275 390-17652 396-17974 401-18266 402-18302 403-18341 407-18577 425-19481 425-19503
			426-19526
WRD.2	=	006302	#85-2243 242-9979 242-9996 244-10092 249-10355 249-10372 251-10468 257-10760 257-10777
			259-10873 264-11129 265-11152
WRD.20	=	006514	#86-2265 252-10522 260-10927 302-13045
WRD.21	=	006522	#86-2266 302-13027 302-13030 302-13042
WRD.22	=	006530	#86-2267 328-14405 348-15420 372-16739
WRD.23	=	006544	#86-2268 372-16735 373-16761 383-17277 402-18275 402-18307 403-18346 407-18579 440-20286
			441-20325
WRD.24	=	006552	#86-2269 322-14040 322-14057 322-14078 323-14099 323-14116 379-17064 383-17281 449-20778
WRD.25	=	006560	#86-2270 379-17063 383-17280
WRD.26	=	006574	#86-2271 437-20114
WRD.27	=	006602	#86-2272 429-19741 431-19833 431-19845 431-19848
WRD.29	=	006610	#86-2273 429-19739
WRD.3	=	006314	#85-2244 237-9731 242-10013 249-10389 257-10794 265-11170 288-12367 293-12611 298-12829
WRD.30	=	006624	#86-2274 434-19991 434-19994
WRD.31	=	006632	#86-2275 430-19764 431-19807
WRD.32	=	006646	#86-2276 322-14043 322-14060 322-14081 323-14102 323-14119
WRD.33	=	006654	#86-2277 322-14041 322-14058 322-14079 323-14100 323-14117
WRD.34	=	006660	#86-2278 322-14045 322-14062 322-14083 323-14104 323-14121

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
WRD.35	=	006666	#86-2279 308-13348 315-13690 334-14721 519-24297
WRD.36	=	006674	#86-2280 315-13689
WRD.37	=	006700	#86-2281 199-7805 200-7863 223-9005 334-14720 364-16283 366-16406 439-20250
WRD.38	=	006710	#86-2282 223-9006
WRD.39	=	006716	#86-2283 366-16408 402-18274 402-18306 403-18345 407-18576
WRD.4	=	006322	#85-2245 237-9748 243-10034 250-10410 258-10815 265-11188 289-12416 290-12462 294-12631
			298-12850
WRD.40	=	006726	#86-2284 309-13422 310-13438
WRD.41	=	006732	#86-2285 419-19222
WRD.42	=	006746	#86-2286 309-13423 310-13439
WRD.43	=	006754	#86-2287 242-9977 242-9994 249-10353 249-10370 257-10758 257-10775 264-11127 265-11150
WRD.44	=	006764	#85-2288
WRD.45	=	006772	#86-2289 308-13349 328-14402
WRD.46	=	007004	#86-2290 309-13425 310-13441 419-19221
WRD.47	=	007016	#86-2291 310-13440
WRD.48	=	007026	#86-2292 347-15412
WRD.49	=	007036	#86-2293
WRD.5	=	006330	#85-2246 229-9338 322-14080 323-14101 323-14118 454-21023 458-21262
WRD.50	=	007046	#86-2294 355-15846 390-17656 396-17978 412-18837 413-18902
WRD.51	=	007054	#86-2295 309-13424 355-15845
WRD.52	=	007066	#86-2296 193-7499 204-8074 210-8327 215-8621 271-11450 272-11514 273-11591 497-23159
			498-23215 504-23527
WRD.53	=	007074	#86-2297 230-9393
WRD.54	=	007104	#86-2298 309-13426
WRD.55	=	007114	#86-2299 309-13427
WRD.56	=	007122	#86-2300 204-8073 210-8326 215-8619 271-11449 272-11513 273-11589 353-15758 364-16282
			412-18835
WRD.57	=	007130	#86-2301
WRD.58	=	007140	#86-2302
WRD.59	=	007152	#86-2303 426-19529
WRD.6	=	006336	#85-2247 230-9362 230-9382 322-14042 322-14059 440-20285 441-20324
WRD.60	=	007156	#86-2304 412-18836 413-18901
WRD.61	=	007174	#86-2305 252-10523 260-10928 464-21548 466-21646 471-21925 478-22219
WRD.62	=	007210	#86-2306 193-7502
WRD.63	=	007216	#87-2311
WRD.64	=	007224	#87-2312 449-20779 454-21022 458-21261
WRD.65	=	007232	#87-2313
WRD.67	=	007240	#87-2314 466-21642 471-21900 471-21921 477-22211 497-23146 511-23889
WRD.68	=	007246	#87-2315
WRD.69	=	007252	#87-2316 497-23144 511-23887
WRD.7	=	006344	#85-2248 229-9337 230-9361 230-9381 230-9394 439-20251 440-20284 440-20319
WRD.70	=	007260	#87-2317
WRD.71	=	007272	#87-2318 507-23686
WRD.72	=	007302	#87-2319 516-24130
WRD.73	=	007310	#87-2320 449-20777
WRD.74	=	007316	#87-2321 464-21524 464-21547 466-21618 466-21645 471-21924 477-22214
WRD.75	=	007330	#87-2322 464-21523 464-21546 465-21613 466-21644 471-21923 477-22213
WRD.76	=	007340	#87-2323 471-21898 477-22188 484-22554 491-22855
WRD.77	=	007346	#87-2324 477-22190
WRD.8	=	006360	#85-2249 230-9380 440-20318 454-21021 458-21260
WRD.9	=	006374	#85-2250 229-9336 230-9360 372-16737 373-16763 440-20283 471-21897 477-22187 484-22553
			491-22854

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
WRT.CS		017462	#123-4082 173-6508 220-8875 221-8907
WRT.DA		020016	#129-4397 173-6518 220-8881 221-8915
WRT.DS		022764	#168-6268 174-6562
WRT.D1		021570	#155-5668 174-6567
WRT.D2		022102	#159-5858 174-6572
WRT.D3		022414	#163-6060 174-6577
WRT.EE		021526	#152-5542 173-6548
WRT.EL		021464	#149-5407 174-6557
WRT.ER		017640	#126-4239 173-6513 220-8878 221-8911
WRT.E1		020560	#138-4870 173-6528 221-8891 221-8923
WRT.E2		020766	#141-5043 173-6533 221-8894
WRT.MR		020174	#132-4554 173-6573
WRT.PA		020352	#135-4710 173-6538 221-8888 221-8919
WRT.PD		021220	#145-5228 173-6543 418-19147
WRT.RE		023074	#172-6481 204-8039 209-8288 214-8562 271-11437 272-11501 273-11569
WT.DAT		013126	#79-1896 *125-4191 125-4195 *128-4349 128-4353 *131-4506 131-4510 *134-4664 134-4668 *137-4823 137-4827 *140-4983 140-4987 *144-5167 144-5171 *148-5353 148-5357 *151-5491 151-5493 *154-5624 154-5626 *158-5809 158-5811 *162-5997 162-5999 *167-6208 167-6212 *170-6366 170-6370 204-8084 210-8337 223-9012 271-11460 272-11524
W.C.SI		013122	#79-1893 *186-7189 *186-7209 411-18798 411-18807 413-18887
XOR.LN		016642	#117-3796 372-16708 379-17053
XSALWA	=	000000	#4-13
XSALS	=	000040	#4-13
XSOFFS	=	000400	#4-13
XSTRUE	=	000020	#4-13
\$END.L		105134	G #523-24451
\$PATCH		002756	G #5-327
\$SAVE2		004044	G 34-128 39-371 #42-61 102-3082 113-3570 123-4082 125-4179 126-4239 128-4337 129-4397 131-4494 132-4554 134-4652 135-4710 137-4810 138-4870 140-4970 141-5043 144-5153 145-5228 148-5339 155-5668 157-5781 159-5858 161-5969 163-6060 166-6179 170-6354 172-6481 176-6679 236-9681 241-9939 248-10315 256-10720 263-11079 301-12987 411-18785 443-20453 519-24273 523-24471
\$SAVE3		004060	G 9-143 10-191 15-153 16-207 28-183 29-234 #42-68 88-2407 100-2975 104-3184 185-7119 226-9144 229-9318 233-9506 281-11971 288-12332 424-19426 510-23837 515-24062 523-24470
\$SAVE4		004076	G #42-76 117-3797 196-7624 203-8019 277-11747 429-19713 523-24470
\$SAVE5		004116	G 35-196 39-371 #42-85 93-2648 97-2832 193-7475 199-7786 209-8271 214-8545 220-8863 270-11417 297-12778 306-13254 313-13604 320-13956 327-14305 332-14598 339-14973 346-15308 352-15638 361-16133 370-16607 377-16953 382-17228 387-17512 394-17865 401-18223 406-18519 417-19098 447-20675 453-20964 457-21208 463-21477 470-21832 476-22118 482-22421 489-22745 495-23041 504-23501 523-24470
\$T1		025670	#189-7332 190-7391
\$T10		032366	#229-9318 231-9422
\$T11		033030	#233-9506 234-9580
\$T12		033274	#236-9681 238-9786
\$T13		033704	#241-9939 245-10160
\$T14		035056	#248-10315 252-10554
\$T15		036324	#256-10720 260-10959
\$T16		037572	#263-11079 266-11244
\$T17		040472	#270-11417 274-11647
\$T18		041666	#277-11747 278-11816
\$T19		042124	#281-11971 285-12200

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
\$T2		026072	#193-7475 194-7547
\$T20		043274	#288-12332 291-12501
\$T21		044234	#293-12579 294-12671
\$T22		044630	#297-12778 299-12893
\$T23		045276	#301-12987 303-13080
\$T24		045634	#306-13254 311-13493
\$T25		047010	#313-13604 316-13742
\$T26		047566	#320-13956 324-14172
\$T27		050712	#327-14305 329-14437
\$T28		051470	#332-14598 335-14769
\$T29		052454	#339-14973 342-15172
\$T3		026336	#196-7624 197-7685
\$T30		053566	#346-15308 348-15448
\$T31		054362	#352-15638 356-15897
\$T32		055756	#361-16133 367-16450
\$T33		057514	#370-16607 373-16789
\$T34		060526	#377-16953 380-17122
\$T35		061462	#382-17228 384-17338
\$T36		062114	#387-17512 391-17716
\$T37		063152	#394-17865 397-18029
\$T38		064060	#401-18223 404-18407
\$T39		065064	#406-18519 408-18631
\$T4		026544	#199-7786 201-7909
\$T40		065512	#411-18785 414-18940
\$T41		066500	#417-19098 420-19280
\$T42		067532	#424-19426 426-19560
\$T43		070304	#429-19713 432-19885
\$T44		071240	#434-19971 435-20028
\$T45		071450	#436-20090 437-20138
\$T46		071634	#439-20247 441-20361
\$T47		072332	#443-20453 444-20520
\$T48		072624	#447-20675 450-20832
\$T49		073516	#453-20964 455-21087
\$T5		027236	#203-8019 205-8135
\$T50		074222	#457-21208 459-21316
\$T51		074640	#463-21477 467-21690
\$T52		075712	#470-21832 472-21973
\$T53		076464	#476-22118 478-22264
\$T54		077264	#482-22421 485-22605
\$T55		100306	#489-22745 491-22898
\$T56		101156	#495-23041 498-23247
\$T57		102260	#504-23501 508-23730
\$T58		103502	#510-23837 512-23932
\$T59		104106	#515-24062 517-24174
\$T6		027660	#209-8271 211-8392
\$T60		104554	#519-24273 521-24365
\$T7		030316	#214-8545 217-8702
\$T8		031146	#220-8863 224-9067