

ML11

LOGIC TEST
CZMLADO

AH-S390D-MC
FICHE 1 OF 3

JUL 1982
COPYRIGHT © 81-82
MADE IN USA



A large grid of approximately 15 columns and 20 rows of small, illegible text and diagrams, likely representing logic test data or circuit diagrams. The content is too faint to transcribe accurately.

ML11

LOGIC TEST
CZMLADO

AH-S390D-MC
FICHE 2 OF 3

JUL 1982
COPYRIGHT © 81-82
MADE IN USA



A large grid of approximately 20 columns and 20 rows of small, illegible text blocks, likely representing a logic test or data table. The text is too small to be read accurately.

ML11

LOGIC TEST
CZMLADO

AH-S390D-MC
FICHE 3 OF 3

JUL 1982
COPYRIGHT © 81-82
MADE IN USA



The main body of the document is a large grid of data, likely a logic test matrix. It consists of approximately 15 columns and 25 rows of small, dense text. Each cell in the grid contains a small amount of information, possibly test results or component identifiers. The text is too small to be legible in this scan, but the overall structure is a regular grid.

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	ASSUMPTIONS
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 THE ML-11.

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

THE FUNCTIONAL LEVEL (FRU) OF THIS DIAGNOSTIC PRODUCT WILL BE TO THE LOGIC FUNCTION LEVEL. 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

PDP-11 WITH MINIMUM OF 28K WORDS OF MEMORY

CONSOLE TERMINAL

RH11 OR RH70

1 TO 8 ML-11A OR ML-11B 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 ASSUMPTIONS

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.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER ^C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

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

2.2 SWITCHES

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

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 6400)
/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 6400)

/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 OF ARRAYS
GPRML	MSGH5,10,1,YES	:DRIVE OPTION TYPE
GPRMD	MSGH6,12,0,7,0,7,YES	:DRIVE NUMBER
GPRML	MSGH7,14,1,YES	:PARITY DISABLED

PARAMETER CODING MESSAGES

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

SAMPLE DIALOGUE

DR> S,A <CR>

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

2.5 SOFTWARE QUESTIONS

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

THE FOLLOWING OPTIONS ARE AVAILABLE FOR OPERATOR SELECTION:

1. ENABLE PRINTING OF DRIVE SERIAL NUMBERS.
2. LIMIT THE NUMBER OF ERRORS PRINTED PER EACH DETECTED ERROR. A MAXIMUM OF 10 ERRORS WILL BE PRINTED PER ERROR BEFORE THE ERROR IS EXITED.
3. ENABLE RH AND ML11 REGISTER DUMP ON ERRORS.
4. ABORTING PROGRAM EXECUTION AFTER ONE COMPLETE PASS.
5. ENABLE MANUAL INTERVENTION TESTS.

2.6 EXTENDED P-TABLE DIALOGUE

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

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT. THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

UNITS (D) ? 8<CR>

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

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

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

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

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

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

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

UNIT 8
CSR ADDRESS (O) 160000<CR>
SUB-DEVICE # (O) ? 7<CR>
Q-FACTOR (O) 1 ? <CR>

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

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

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

UNITS (D) ? 8<CR>

UNIT 1
CSR ADDRESS (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 (D) ? 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
3. TYPE 'R CZMLA'
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH 'Y'

6. ANSWER ALL THE HARDWARE QUESTIONS

7. ANSWER THE "CHANGE SW" QUESTION WITH 'N'

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

ERROR MESSAGES ARE HANDLED VIA A DICTIONARY STRUCTURE.

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

THIS PERMITS EXTENSIVE ERROR MESSAGE PRINTING AT MINIMAL STORAGE REQUIREMENTS.

THE FOLLOWING DEMONSTRATES TYPICAL ERROR MESSAGES:

ML11 DVC FTL ERR 00077 ON UNIT 07 TST 027 SUB 002 PC: 050432
ASYNCHRONOUS MODULE FAILURE
EXCESSIVE DATA ERRORS DURING INITIAL ARR RD_WRT

ML11 DVC FTL ERR 00112 ON UNIT 03 TST 037 SUB 000 PC: 056466
ASYNCHRONOUS MODULE FAILURE

ARRAY ADRS MULTIPLEXER FAILURE
FAILED AT DSA: 000000

3.2.1 ERROR NUMBER DEFINITION

ERROR NO. -----	FAILING LOGIC -----
1	DRIVE DID NOT RESPOND WITHIN 1.5 US
2	DSA REG READ/WRITE ERROR DURING DRIVE SEL TEST
3	UNIQUE DRIVE SELECTION ERROR
4	ML REGISTER READ WRITE ONES/ZEROES ERROR
5	ML REGISTER SHIFTING ONES/ZEROES ERROR
6	ML REGISTER INITIALIZATION ERROR
7	CONTROL BUS BAD PARITY NOT DETECTED. (CPAR NOT SET)
8	CONTROL BUS GOOD PARITY NOT DETECTED. (CPAR SET)
9	CONTROL BUS BAD PARITY GENERATED. (MCPE SET)
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 CLEARED WHILE GO CLEARED DURING WRITE CHECK FUNCTION TEST
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
49 ERROR REGISTER DID NOT CLEAR AFTER DRIVE CLEAR.
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 ATTN BIT NOT SET AFTER SEARCH FUNCTION
60 OPI BIT SET DURING SEARCH FUNCTION AT PRESENT ARRAYS
61 OPI BIT NOT SET AFTER SEARCH AT NON-PRESENT ARRAYS
62 GO BIT NOT CLEAR AFTER READ-IN-PRESET FUNCTION
63 ILF BIT SET DURING READ-IN-PRESET FUNCTION
64 OPI BIT SET DURING READ-IN-PRESET FUNCTION
65 ECC INITIALIZE DID NOT CLEAR MEMORY AFTER POWER UP WITH BAD BATTERY BACK-UP
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 INTERMEDIATE FAILURE. SOME BITS IN NIBBLES TESTED WERE IN ERROR (CONTINUE TESTING)
- 85 SYNC BUS DATA BIT WRITE PATH CONTINUITY FAILURE
- 86 SYNC BUS DATA BIT READ PATH CONTINUITY FAILURE
- 87 RAM BUS ADRS COUNTER FAILURE TO LOAD/UNLOAD SKIP RAM DURING WRITE FUNCTION
- 88 RAM BUS ADRS COUNTER FAILURE TO LOAD/UNLOAD SKIP RAM DURING READ FUNCTION
- 89 SYNC DATA BUS WRITE PATH UNIQUE DATA BIT FAILURE (ALL ONES NIBBLE PATTERN)
- 90 SYNC DATA BUS WRITE PATH UNIQUE DATA BIT FAILURE (SHIFTED BIT NIBBLE PATTERN)
- 91 SYNC DATA BUS WRITE PATH UNIQUE DATA BIT FAILURE (ALL ONES NIBBLE PATTERN)
- 92 SYNC DATA BUS READ PATH UNIQUE DATA BIT FAILURE
- 93 NIBBLE OFF SET COUNTERS FAILURE
- 94 CS1 FUNCTION ABORT FAILURE DURING CLASS 'A' ERROR
- 95 CS1 FUNCTION ABORT FAILURE DURING CLASS 'B' ERROR
- 96 LBT BIT SET BEFORE A LAST BLOCK TRANSFER
- 97 DSA REGISTER INCREMENT FAILURE DURING NON LAST BLOCK TRANSFERS.
- 98 LBT BIT NOT CLEAR AFTER LOADING DSA REG
- 100 LBT BIT NOT SET AFTER A LAST BLOCK TRANSFER
- 101 DSA REGISTER INCREMENT FAILURE AFTER A LAST BLOCK TRANSFER
- 102 IAE BIT NOT SET AT INVALID SECTOR ADDRESSES
- 103 AOE BIT NOT SET AFTER ADDRESS OVERFLOW
- 104 SC BIT NOT AFTER CS1 FUNCTION ADORT
- 105 GOOD DATA BUS PARITY NOT DETECTED
- 106 GOOD DATA BUS PARITY NOT GENERATED
- 107 UNS BIT SET AFTER WRITING TO A SECTOR DURING PROM DATA TEST
- 108 UNS BIT NOT SET WITH BAD UV DATA
- 109 WCE BIT SET DURING MBUS WRITE/READ FUNCTION TROUBLE SHOOTING LOOP TEST
- 110 UNIQUE REGISTER SELECTION TEST FAILURE

- 111 FAILURE TO FIND GOOD MOS RAM ROW DURING ARRAY ADRS MUX TEST (INTERMEDIATE DIAG MSG)
- 112 UNIQUE ARRAY MODULE ROW/COL ADDRESSING FAILURE
- 113 DRIVE TYPE REGISTER VALUE WAS NOT CORRECT
- 114 TRE BIT SET UNEXPECTEDLY DURING A WRITE CHECK TRANSFER (INTERMEDIATE DIAG ERROR)
- 115 TRE BIT SET UNEXPECTEDLY DURING A WRITE TRANSFER (INTERMEDIATE DIAG ERROR)
- 116 TRE BIT SET UNEXPECTEDLY DURING A READ TRANSFER (INTERMEDIATE DIAG ERROR)
- 117 TRE BIT DID NOT SET AFTER A REGISTER MODIFICATION ERROR (EXCEPTION WAS NOT ASSERTED)
- 118 DATA DIAGNOSTIC REGISTER (D1 D2 E2) INITIALIZATION ERRORS
- 119 RH CONTROLLER FAILED EXISTANCE PROBE TEST
- 120 NED BIT SET DURING MASS BUS HANDSHAKE PROBE
- 121 DATA DIAGNOSTIC REGISTER (D1 D2 E2) ONE'S ZERO'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 B
- 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
- 142 FAILURE TO FIND GOOD BLOCK DURING ARRAY SELECT TEST
- 148 SC BIT SET DURING MASS BUS WRITE OR WRITE CHECK TRANSFERS DURING TEST 21 AND 61
- 149 VV BIT WAS SET WHEN TESTED AND GOT CLEARED AFTER A READ-IN-PRESET FUNCTION WAS PERFO
- 150 VV BIT WAS CLEARED WHEN TESTED AND DID NOT GET SET WHEN A READ-IN-PRESET FUNCTION WA
- 151 THE UNS BIT DID NOT SET WHEN BOX POWER WAS LOST
- 152 THE VV BIT DID NOT CLEAR WHEN DRIVE AC POWER RESUMED WITH BAD BATTERY BACK-UP
- 153 DC LO DID NOT INITIATE 'MB DIS LO' RESULTING IN THE ML-11 REGISTERS NOT GETTING CLEA
POWER UP.
- 154 VV BIT GOT CLEARED AFTER POWER FAIL WITH GOOD BATTERY BACK-UP
- 155 ECC INIT CLEARED OUT MEMORY AFTER POWER FAILED WITH GOOD BATTERY BACK-UP
- 156 BATTERY BACK-UP FAILED TO MAINTAIN DATA INTEGRITY DURING PCWER FAIL

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 DEVICE INFORMATION TABLES

HARDWARE DEFAULT PTABLE

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

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. POWER FAIL TEST *MANUAL INTERVENTION TEST*
TEST THE VV BIT SETTING/CLEARING, ECC INIT, REGISTER CLEARING
UNS BIT SETTING, AC AND DC LOW.

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 READ-
ING 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 SIGLE 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.
- TST61. BATTERY BACK-UP TEST *MANUAL INTERVENTION TEST*
TEST THE DRIVE BATTERY BACK-UP DURING POWER FAILS AND ALSO TEST VV BIT, ECC INIT.

```
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 ALL
47
64
65 002000          HEADER CZMLA,D,0,1800.,0
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          ERRTYP::          .WORD 0
98 002154 000000          ERRNBR::          .WORD 0
99 002156 000000          ERRMSG::          .WORD 0
100 002160 000000          ERRBLK::          .WORD 0
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 002162          DISPATCH 61
108
109
110
111          :++
112          : THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
113          : THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
114          : IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
115          : AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
116          :--
117
118
```

```
119 002356          BGNHW  DFPTBL
120
130
131 002360 176400   .WORD 176400   :RH ADDRESS
132 002362 000070   .WORD 70       :RH TYPE
133 002364 000204   .WORD 204      :RH VECTOR ADDRESS
134 002366 000001   .WORD 1        :NUMBER OF ARRAY MODULES
135 002370 000001   .WORD 1        :IS DRIVE OPTION ML11A, 1=16K, 0=64K
136 002372 000000   .WORD 0        :ML-11 DRIVE NUMBER
137 002374 000000   .WORD 0        :IS PARITY DISABLED, 1=YES, 0=NO
```

```
138
139 002376          ENDSW
```

```
140
141
142
143
144 :++
145 : THE DEFAULT SOFTWARE P-TABLE CONTAINS VARIOUS DATA USED BY THE
146 : PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE SET
147 : UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR AT RUN
148 : TIME.
149 :--
```

```
150 002376          BGNSW  SFPTBL
151
159
160 002400 000000   PRSN::        .WORD 0        :PRINT SERIAL NUMBER, 1=YES, 0=NO
161 002402 000001   ERRTHR::      .WORD 1        :ENABLE ERROR THRESHOLD 1=YES, 0=NO
162 002404 000000   REGDMP::      .WORD 0        :ENABLE REGISTER DUMPS 1=YES, 0=NO
163 002406 000001   ONEPAS::      .WORD 1        :DROP UNIT AFTER ONE (EOP)
164 002410 000000   INTERVEN::    .WORD 0        :MANUAL INTERVENTION TESTS 1=YES, 0=NO
165 002412
166          ENDSW
```


192
218
219
220
221
222
223
224
225
226
227
228
229 002412
230
240
241
242 002414
243 002424
244 002436
245 002450
246 002462
247 002470
248 002502
249
250 002510
251
258
259 002510
260 002556
261 002624
262 002670
263 002736
264 003004
265 003052
266
267
268
269
270
271
272
273
274
275
276
277 003120
278
287
288 003122
289 003130
290 003136
291 003144
292 003152
293
294
295 003160
296

```

:++
: THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
: THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
: MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
: INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
: MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
: WITH THE OPERATOR.
:--

```

BGNHRD

```

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

```

ENDHRD

```

MSGH1: .ASCIZ /RH CONTROLER BASE REGISTER ADDRESS ?/
MSGH2: .ASCIZ /RH CONTROLER TYPE '70 OR 11' ?/
MSGH3: .ASCIZ /RH CONTROLER VECTOR ADDRESS ?/
MSGH4: .ASCIZ /NUMBER OF ARRAY MODULES IN THIS UUT ?/
MSGH5: .ASCIZ /IS UUT DRIVE OPTION TYPE AN 'ML11A' ?/
MSGH6: .ASCIZ /DRIVE SELECT NUMBER OF THIS UUT ?/
MSGH7: .ASCIZ /IS PARITY DISABLED IN THIS UUT ?/
.EVEN

```

```

:++
: 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
GPRML MSGS2,2,1,YES
GPRML MSGS3,4,1,YES
GPRML MSGS4,6,1,YES
GPRML MSGS5,10,1,YES
.EVEN

```

ENDSFT

```
297
304 003160      120      122      111  MSGS1:  .ASCIZ  /PRINT DRIVE SERIAL NUMBER      ?/      :PRINT DRIVE SERIAL NUMBER?
305 003224      105      116      101  MSGS2:  .ASCIZ  /ENABLE ERROR MESSAGE THRESHOLD  ?/      :ERROR THRESHOLD FLAG
306 003270      105      116      101  MSGS3:  .ASCIZ  /ENABLE REGISTER DUMP ON ERRORS    ?/      :REG DUMP FLAG
307 003334      105      130      111  MSGS4:  .ASCIZ  /EXIT PROGRAM AFTER ONE PASS      ?/      :EXIT PROG FLAG
308 003400      105      116      101  MSGS5:  .ASCIZ  /ENABLE MANUAL INTERVENTION TESTS ?/
309           .EVEN
310
311
312           :++
313           : THIS TABLE IS USED BY THE RUNTIME SERVICES
314           : TO PROTECT THE LOAD MEDIA.
315           :--
316 003444           BGNPROT
317
318 003444      177777      -1           :OFFSET INTO P-TABLE FOR CSR ADDRESS
319 003446      177777      -1           :OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
320 003450      177777      -1           :OFFSET INTO P-TABLE FOR DRIVE NUMBER
321
322 003452           ENDPROT
323
337
338 003452           $PATCH::
339 003452           .BLKW  20
340
347
348 003512           ENDMOD
349
350
363
```

22-Mar-1982 11:35:31
22-Mar-1982 11:32:50

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML3AD.BLI.1 (1)

```

6 :ML3AD
7 :
8 :
9 :      0001  MODULE ML3AD =
10 :      0002  BEGIN
11 :      0003
12 :      0004  %SBTTL 'REPORT CODE SECTION'
13 :      0005
14 :      0006  REQUIRE 'BLSMAC.REQ';
15 :      1496
16 :      1497  !+
17 :      1498  ! THE REPORT CODING SECTION CONTAINS THE
18 :      1499  ! 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
19 :      1500  !-
20 :      1501
21 :      1502  BGNRPT;
22 :      1503  RETURN;
23 :      1504  ENDRPT;

```

```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

          .SBTTL  LRPT REPORT CODE SECTION
LRPT:  RTS    PC
:
: Routine Size: 1 word
: Maximum stack depth per invocation: 0 words

```

1495

```

52 004650 004767 177772
53 004654 104425
54 004656 000207

```

```

          .SBTTL  LSRPT REPORT CODE SECTION
LSRPT:: JSR    PC,LRPT
        TRAP   25
        RTS    PC
:
: Routine Size: 4 words
: Maximum stack depth per invocation: 0 words

```

1503

22-Mar-1982 11:35:31
22-Mar-1982 11:32:50

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML3AD.BLI.1 (1)

116 :ML3AD

REPORT CODE SECTION

117 :
118 :
119 : 1516
120 : 1517
121 : 1518
122 : 1519
123 : 1520
124 : 1521
125 : 1522
126 : 1523
127 : 1524

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

BGNDU;
RETURN;
ENDDU;

131
132
136 004672 000207

.SBTTL LDU REPORT CODE SECTION
LDU: RTS PC ;

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

1515

137
138
139
144
145
149
150
154 004674 004767 177772
155 004700 104453
156 004702 000207

.SBTTL LSDU REPORT CODE SECTION
LSDU:: JSR PC,LDU ;
TRAP 53
RTS PC

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

1523

157
158
159
164
165
166 : 1525
167 : 1526
168 : 1527
169 : 1528
170 : 1529

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

22-Mar-1982 11:35:31 TOPS-20 Bliss-16 V2(212)
22-Mar-1982 11:32:50 PA:<NEALE>ML3AD.BLI.1 (1)

172 :ML3AD
173 :
174 :
175 : 1530 !-
176 : 1531
177 : 1532 BGNAU;
178 : 1533 RETURN;
179 : 1534 ENDAU;
183
184

REPORT CODE SECTION

188 004704 000207

LAU: .SBTTL LAU REPORT CODE SECTION
RTS PC ;

1524

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

190
191
196
197
201
202

206 004706 004767 177772
207 004712 104452
208 004714 000207

LSAU:: .SBTTL LSAU REPORT CODE SECTION
JSR PC,LAU ;
TRAP 52
RTS PC

1533

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

209
210
211
216
217
218 : 1535 END
219 : 1536
220 : 1537 ELUDOM
224
225
226

228
229
230
231
232
233
234
235
236
237
238

:ML3AD
:
REPORT CODE SECTION

: Size: 20 code + 0 data words
: Run Time: 00:02.9
: Elapsed Time: 00:05.5
: Memory Used: 24 pages
: Compilation Complete

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (1)

```

6 :ML4AD
7 :
8 :
9 :      0001 MODULE ML4AD =
10 :      0002 BEGIN
11 :      0003 :
12 :      0004 : PRETTY BLF COMMANDS
13 :      0005 :
14 :      0006 : <BLF/LOWERCASE_KEY>
15 :      0007 :
16 :      0008 : REQUIRE
17 :      0009 :
18 :      0010 :
19 :      0011 %SBTTL 'DECLARATION SECTION'
20 :      0012 :
21 :      0013 require 'BLSMAC.REQ';
22 :      1503 :
23 :      1504 :
24 :      1505 : CONSTANT LITERALS
25 :      1506 :
26 :      1507 :
27 :      1508 literal
28 :      1509 ONE = 1,
29 :      1510 ONES = %o'177777',
30 :      1511 ZERO = 0,
31 :      1512 ZEROES = 0,
32 :      1513 NUM OF REG = 22,
33 :      1514 FIELD_SIZ = 4,
34 :      1515 :
35 :      1516 : MLCS1 FUNCTION CODES
36 :      1517 :
37 :      1518 NOOP = 1,
38 :      1519 DRV CLR = %o'11',
39 :      1520 RD IN PRE = %o'21',
40 :      1521 SEARCH = %o'31',
41 :      1522 WRT CHK = %o'51',
42 :      1523 write = %o'61',
43 :      1524 read = %o'71',
44 :      1525 :
45 :      1526 : DELAY ARGUMENTS
46 :      1527 :
47 :      1528 ONE_US = 1,
48 :      1529 FRTY_US = 40,
49 :      1530 TWO_TH_US = 2000,
50 :      1531 ONE_SEC = 100,
51 :      1532 ONE_MS = 10;
52 :      1533 :
53 :      1534 :
54 :      1535 : FIELD DECLARATIONS
55 :      1536 :
56 :      1537 :
57 :      1538 field
58 :      1539 WORD_MAP =
59 :      1540 set
60 :      1541 REGISTER_ADD = [0, 0, 16, 0],

```

!BLISS INTERFACE MODULE

!DATA BIT OF ONE
!DATA PATTERN OF ONES
!DATA BIT OF ZERO
!DATA PATTERN OF ZEROES
!NUMBER OF BLOCKS IN GLOBAL STORAGE 'ML-REG'
!FIELD SIZ FOR FIELD DECLARATIONS 'WORD_MAP'

!NOOP FUNCTION
!DRIVE CLEAR FUNCTION
!READ IN PRESET FUNCTION
!SEARCH FUNCTION
!WRITE CHECK FUNCTION
!WRITE FUNCTION
!READ FUNCTION

!ONE MICRO SECOND DELAY
!FORTY MICRO SECOND DELAY
!TWO THOUSAND MICRO SECOND DELAY
!ONE SECOND DELAY
!ONE MILL SECOND DELAY

!MAPS GLOBAL STORAGE 'ML-REG' INTO REGISTER PERSONALITIES
!REGISTERS ADDRESS

62 : ML4AD
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 :
94 :
95 :
96 :
97 :
98 :
99 :
100 :
101 :
102 :
103 :

DECLARATION SECTION

```
1542 FORCE_HI = [1, 0, 16, 0],
1543 FORCE_LO = [2, 0, 16, 0],
1544 DONT_CARE = [3, 0, 16, 0],
1545 tes,
1546 NIB_MAP =
1547 set
1548 NIB_0 = [0, 0, 4, 0],
1549 NIB_1 = [0, 4, 4, 0],
1550 NIB_2 = [0, 8, 4, 0],
1551 NIB_3 = [0, 12, 4, 0],
1552 NIB_4 = [1, 0, 4, 0],
1553 NIB_5 = [1, 4, 4, 0],
1554 NIB_6 = [1, 8, 4, 0],
1555 NIB_7 = [1, 12, 4, 0],
1556 NIB_8 = [2, 8, 4, 0],
1557 NIB_9 = [2, 12, 3, 0],
1558 tes,
1559 DT1_FLD =
1560 set
1561 EO_5 = [0, 0, 6, 0],
1562 CO_5 = [0, 6, 6, 0],
1563 CRC_DATA = [0, 13, 1, 0],
1564 SGL_DATA = [0, 14, 1, 0],
1565 UNC_DATA = [0, 15, 1, 0],
1566 EE_DATA = [0, 0, 16, 0],
1567 AO_5 = [1, 0, 6, 0],
1568 PO_5 = [1, 6, 6, 0],
1569 ECH_DATA = [1, 12, 1, 0],
1570 BO_5 = [2, 0, 6, 0],
1571 tes;
1572
1573 external
1574 REGDMP : volatile,
1575 ONEPAS : volatile,
1576 INTERVEN : volatile,
1577 PRSN : volatile,
1578 ERRTHR : volatile;
1579
1580 !<BLF/PAGE>
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (1)

!REGISTERS FORCED HI BITS
!REGISTERS FORCED LO BITS
!REGISTERS IGNORE BITS

!MAPS OWN STORAGE NIB_SAVE INTO TEN FOUR BIT NIBBLES

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

!REGISTER DUMP ON ERROR FLAG
!DROP UNIT AFTER ONE PASS
!MANUAL INTERVENTION FLAG
!PRINT SERIAL NUMBER FLAG
!ENABLE ERROR THRESHOLD FLAG

105 :ML4AD

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (2)

DECLARATION SECTION

```

106 :
107 :
108 : 1581 :
109 : 1582 : OWN STORAGE
110 : 1583 :
111 : 1584 :
112 : 1585 : own
113 : 1586 : NIB_SAVE : block [3] field (NIB_MAP) volatile,
114 : 1587 :
115 : 1588 : HW OR TBL : vector [127] volatile,
116 : 1589 : PTBL_PTR : volatile,
117 : 1590 : OP_NUM_ARR : volatile,
118 : 1591 : ARR_INC : volatile,
119 : 1592 : GOOD_BLK : volatile,
120 : 1593 : PAR_DIS : volatile,
121 : 1594 : CHIP_SIZ : volatile,
122 : 1595 : LST_BLK : volatile,
123 : 1596 : ARR_16 : volatile,
124 : 1597 : LST_ARR : volatile,
125 : 1598 : IO_BUF : vector [256] volatile,
126 : 1599 : STR_OFF : vector [10, byte] volatile,
127 : 1600 : stack : vector [198, byte] volatile,
128 : 1601 : PD_TEMP : bitvector [16] volatile,
129 : 1602 : WT_SIZE : volatile,
130 : 1603 : RAS_INC : volatile,
131 : 1604 : WT_DATA : volatile,
132 : 1605 : RD_DATA : volatile,
133 : 1606 : DRIVE_TYPE : volatile,
134 : 1607 : LST_DOT_REG : volatile,
135 : 1608 : REG_INIT_FLG : initial (0) volatile,
136 : 1609 : A_CAL : volatile,
137 : 1610 : B_CAL : volatile,
138 : 1611 : P_CAL : volatile,
139 : 1612 : A_GEN : volatile,
140 : 1613 : B_GEN : volatile,
141 : 1614 : P_GEN : volatile,
142 : 1615 : P_CNT : volatile,
143 : 1616 : LIMIT : volatile,
144 : 1617 : DROP_CNT : volatile,
145 : 1618 :
146 : 1619 : <BLF/NOFORMAT>
147 : 1620 :
148 : 1621 : ML_REG: blockvector [NUM_OF_REG, FIELD_SIZ] field(WORD_MAP) !ML11 REGISTERS
149 : 1622 : preset (
150 : 1623 :
151 : 1624 :
152 : 1625 :
153 : 1626 :
154 : 1627 :
155 : 1628 : [0, FORCE_HI] = %o'004000', !MLCS1
156 : 1629 :
157 : 1630 : [0, FORCE_LO] = %o'173701',
158 : 1631 : [0, DONT_CARE] = %o'160200',
159 : 1632 :

```

```

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

```


29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (2)

161 :ML4AD
162 :
163 :
164 :
165 :
166 :
167 :
168 :
169 :
170 :
171 :
172 :
173 :
174 :
175 :
176 :
177 :
178 :
179 :
180 :
181 :
182 :
183 :
184 :
185 :
186 :
187 :
188 :
189 :
190 :
191 :
192 :
193 :
194 :
195 :
196 :
197 :
198 :
199 :
200 :
201 :
202 :
203 :
204 :
205 :
206 :
207 :
208 :
209 :
210 :
211 :
212 :
213 :
214 :
215 :

DECLARATION SECTION

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

REM_TBL:vector [63,byte]
preset (

[0] = %b'000001'
[1] = %b'000010'
[2] = %b'000100'
[3] = %b'001000'
[4] = %b'010000'
[5] = %b'100000'
[6] = %b'000011'
[7] = %b'000110'
[8] = %b'001100'
[9] = %b'011000'
[10] = %b'110000'
[11] = %b'100011'
[12] = %b'000101'
[13] = %b'001010'
[14] = %b'010100'
[15] = %b'101000'
[16] = %b'010011'
[17] = %b'100110'
[18] = %b'001111'
[19] = %b'011110'
[20] = %b'111100'
[21] = %b'111011'
[22] = %b'110101'
[23] = %b'101001'

!REMAINDER TABLE
!STRUCTURE TO STORE PRECALCULATED
!CRC REMAINDER VALUES FOR CRC CODE
!GENERATION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (2)

217 :ML4AD

DECLARATION SECTION

```
218 :  
219 :  
220 : 1685 [24] = %b'010001'  
221 : 1686 [25] = %b'100010'  
222 : 1687 [26] = %b'000111'  
223 : 1688 [27] = %b'001110'  
224 : 1689 [28] = %b'011100'  
225 : 1690 [29] = %b'111000'  
226 : 1691 [30] = %b'110011'  
227 : 1692 [31] = %b'100101'  
228 : 1693 [32] = %b'001001'  
229 : 1694 [33] = %b'010010'  
230 : 1695 [34] = %b'100100'  
231 : 1696 [35] = %b'001011'  
232 : 1697 [36] = %b'010110'  
233 : 1698 [37] = %b'101100'  
234 : 1699 [38] = %b'011011'  
235 : 1700 [39] = %b'110110'  
236 : 1701 [40] = %b'101111'  
237 : 1702 [41] = %b'011101'  
238 : 1703 [42] = %b'111010'  
239 : 1704 [43] = %b'110111'  
240 : 1705 [44] = %b'101101'  
241 : 1706 [45] = %b'011001'  
242 : 1707 [46] = %b'110010'  
243 : 1708 [47] = %b'100111'  
244 : 1709 [48] = %b'001101'  
245 : 1710 [49] = %b'011010'  
246 : 1711 [50] = %b'110100'  
247 : 1712 [51] = %b'101011'  
248 : 1713 [52] = %b'010101'  
249 : 1714 [53] = %b'101010'  
250 : 1715 [54] = %b'010111'  
251 : 1716 [55] = %b'101110'  
252 : 1717 [56] = %b'011111'  
253 : 1718 [57] = %b'111110'  
254 : 1719 [58] = %b'111111'  
255 : 1720 [59] = %b'111101'  
256 : 1721 [60] = %b'111001'  
257 : 1722 [61] = %b'110001'  
258 : 1723 [62] = %b'100001' volatile,  
259 : 1724  
260 : 1725 DT_1:blockvector[5,3] field (DT1_FLD)  
261 : 1726 preset (  
262 : 1727 [0,E0_5] = %b'111111'  
263 : 1728 [0,C0_5] = %b'100100'  
264 : 1729 [0,CRC_DATA] = %b'1'  
265 : 1730 [0,SGL_DATA] = %b'1'  
266 : 1731 [0,UNC_DATA] = %b'0'  
267 : 1732 [0,A0_5] = %b'000000'  
268 : 1733 [0,P0_5] = %b'111111'  
269 : 1734 [0,ECH_DATA] = %b'0'  
270 : 1735 [0,B0_5] = %b'000000'  
271 : 1736 [1,E0_5] = %b'111111'
```

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (2)

273 :ML4AD
274 :
275 :
276 :
277 :
278 :
279 :
280 :
281 :
282 :
283 :
284 :
285 :
286 :
287 :
288 :
289 :
290 :
291 :
292 :
293 :
294 :
295 :
296 :
297 :
298 :
299 :
300 :
301 :
302 :
303 :
304 :
305 :
306 :
307 :
308 :
309 :
310 :
311 :
312 :
313 :
314 :
315 :
316 :
317 :
318 :
319 :
320 :
321 :
322 :

DECLARATION SECTION

[1.CO_5] = %b'100100'
[1.CRC_DATA] = %b'1'
[1.SGL_DATA] = %b'1'
[1.UNC_DATA] = %b'0'
[1.A0_5] = %b'111111'
[1.P0_5] = %b'000000'
[1.ECH_DATA] = %b'0'
[1.B0_5] = %b'000000'
[2.E0_5] = %b'111111'
[2.CO_5] = %b'000101'
[2.CRC_DATA] = %b'0'
[2.SGL_DATA] = %b'0'
[2.UNC_DATA] = %b'1'
[2.A0_5] = %b'000000'
[2.P0_5] = %b'111111'
[2.ECH_DATA] = %b'1'
[2.B0_5] = %b'111111'
[3.E0_5] = %b'111111'
[3.CO_5] = %b'000101'
[3.CRC_DATA] = %b'0'
[3.SGL_DATA] = %b'0'
[3.UNC_DATA] = %b'1'
[3.A0_5] = %b'000000'
[3.P0_5] = %b'000000'
[3.ECH_DATA] = %b'0'
[3.B0_5] = %b'111111'
[4.E0_5] = %b'111111'
[4.CO_5] = %b'000101'
[4.CRC_DATA] = %b'0'
[4.SGL_DATA] = %b'0'
[4.UNC_DATA] = %b'1'
[4.A0_5] = %b'111111'
[4.P0_5] = %b'000000'
[4.ECH_DATA] = %b'1'
[4.B0_5] = %b'111111')volatile.

!<BLF/FORMAT>

RH_ADD : volatile,
RH_TYP : volatile,
RH_VEC : volatile,
ML_LUN : volatile,
ML_DUT : volatile;

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

!<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

324 :ML4AD

325 :
326 :
327 :
328 :
329 :
330 :
331 :
332 :
333 :
334 :
335 :
336 :
337 :
338 :
339 :
340 :
341 :
342 :
343 :
344 :
345 :
346 :
347 :
348 :
349 :
350 :
351 :
352 :
353 :
354 :
355 :
356 :
357 :
358 :
359 :
360 :
361 :
362 :
363 :
364 :
365 :
366 :
367 :
368 :
369 :
370 :
371 :
372 :
373 :
374 :
375 :
376 :
377 :
378 :

DECLARATION SECTION

EQUALS;

MACRO DEFINITIONS

macro

REGISTER NAMES:

M 1794	MLCS1 =	
1795	.ML_REG [0,REGISTER_ADD]%,	!CONTROL AND STATUS REGISTER 1
M 1796	MLWC =	
1797	.ML_REG [1,REGISTER_ADD]%,	!WORD COUNT REGISTER
M 1798	MLBA =	
1799	.ML_REG [2,REGISTER_ADD]%,	!UNIBUS ADDRESS REGISTER
M 1800	MLDA =	
1801	.ML_REG [3,REGISTER_ADD]%,	!DESIRED ADDRESS REGISTER
M 1802	MLCS2 =	
1803	.ML_REG [4,REGISTER_ADD]%,	!CONTROL AND STATUS REGISTER 2
M 1804	MLDS =	
1805	.ML_REG [5,REGISTER_ADD]%,	!DRIVE STATUS REGISTER
M 1806	MLER =	
1807	.ML_REG [6,REGISTER_ADD]%,	!ERROR REGISTER
M 1808	MLAS =	
1809	.ML_REG [7,REGISTER_ADD]%,	!ATTENTION SUMMARY REGISTER
M 1810	MLLA =	
1811	.ML_REG [8,REGISTER_ADD]%,	!LOOK AHEAD REGISTER
M 1812	MLPA =	
1813	.ML_REG [8,REGISTER_ADD]%,	!PROM ADDRESS REGISTER
M 1814	MLDB =	
1815	.ML_REG [9,REGISTER_ADD]%,	!DATA BUFFER REGISTER
M 1816	MLMR =	
1817	.ML_REG [10,REGISTER_ADD]%,	!MAINTENANCE REGISTER
M 1818	MLDT =	
1819	.ML_REG [11,REGISTER_ADD]%,	!DRIVE TYPE REGISTER
M 1820	MLSN =	
1821	.ML_REG [12,REGISTER_ADD]%,	!SERIAL NUMBER REGISTER
M 1822	MLE1 =	
1823	.ML_REG [13,REGISTER_ADD]%,	!ECC CRC WORD REGISTER 1
M 1824	MLE2 =	
1825	.ML_REG [14,REGISTER_ADD]%,	!ECC CRC WORD REGISTER 2
M 1826	MLD1 =	
1827	.ML_REG [15,REGISTER_ADD]%,	!DATA DIAGNOSTIC REGISTER 1
M 1828	MLD2 =	
1829	.ML_REG [16,REGISTER_ADD]%,	!DATA DIAGNOSTIC REGISTER 2
M 1830	MLEE =	
1831	.ML_REG [17,REGISTER_ADD]%,	!ECC ERROR REGISTER
M 1832	MLEL =	
1833	.ML_REG [18,REGISTER_ADD]%,	!ECC ERROR LOCATION REGISTER
M 1834	MLPD =	
1835	.ML_REG [19,REGISTER_ADD]%,	!PROM DATA REGISTER

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (3)

380 :ML4AD

DECLARATION SECTION

```
381 :  
382 :  
383 : M 1836 MLBAE =  
384 : M 1837 .ML_REG [20,REGISTER_ADD]%,  
385 : M 1838 MLCS3 =  
386 : M 1839 .ML_REG [21,REGISTER_ADD]%,  
387 : M 1840  
388 : M 1841  
389 : M 1842  
390 : M 1843 SC =  
391 : M 1844 (MLCS1)<15,1>%,  
392 : M 1845 TRE =  
393 : M 1846 (MLCS1)<14,1>%,  
394 : M 1847 MCPE =  
395 : M 1848 (MLCS1)<13,1>%,  
396 : M 1849 DVA =  
397 : M 1850 (MLCS1)<11,1>%,  
398 : M 1851 RDY =  
399 : M 1852 (MLCS1)<7,1>%,  
400 : M 1853 IE =  
401 : M 1854 (MLCS1)<6,1>%,  
402 : M 1855 GO =  
403 : M 1856 (MLCS1)<0,1>%,  
404 : M 1857 ML_FUNC =  
405 : M 1858 (MLCS1)<0,6>%,  
406 : M 1859 DLT =  
407 : M 1860 (MLCS2)<15,1>%,  
408 : M 1861 WCE =  
409 : M 1862 (MLCS2)<14,1>%,  
410 : M 1863 PE =  
411 : M 1864 (MLCS2)<13,1>%,  
412 : M 1865 NED =  
413 : M 1866 (MLCS2)<12,1>%,  
414 : M 1867 NEM =  
415 : M 1868 (MLCS2)<11,1>%,  
416 : M 1869 PGE =  
417 : M 1870 (MLCS2)<10,1>%,  
418 : M 1871 MXF =  
419 : M 1872 (MLCS2)<9,1>%,  
420 : M 1873 MDPE =  
421 : M 1874 (MLCS2)<8,1>%,  
422 : M 1875 ORDY =  
423 : M 1876 (MLCS2)<7,1>%,  
424 : M 1877 IRDY =  
425 : M 1878 (MLCS2)<6,1>%,  
426 : M 1879 CLR =  
427 : M 1880 (MLCS2)<5,1>%,  
428 : M 1881 PAT =  
429 : M 1882 (MLCS2)<4,1>%,  
430 : M 1883 BAI =  
431 : M 1884 (MLCS2)<3,1>%,  
432 : M 1885 DRV_NUM =  
433 : M 1886 (MLCS2)<0,3>%,  
434 : M 1887 ATTN =
```

!BUS ADDRESS EXTENSION REGISTER

!CONTROL AND STATUS REGISTER 3

BIT ASSIGNMENTS:

!MLCS1 BIT ASSIGNMENTS

!MLCS2 BIT ASSIGNMENTS

```
436 :ML4AD
437 :
438 :
439 :      1888      (MLDS)<15,1>%,
440 :      M 1889      COMP_ERR =
441 :      1890      (MLDS)>14,1>%,
442 :      M 1891      MOL =
443 :      1892      (MLDS)<12,1>%,
444 :      M 1893      LBT =
445 :      1894      (MLDS)<10,1>%,
446 :      M 1895      DPR =
447 :      1896      (MLDS)<8,1>%,
448 :      M 1897      DRY =
449 :      1898      (MLDS)<7,1>%,
450 :      M 1899      VV =
451 :      1900      (MLDS)<6,1>%,
452 :      M 1901      DCK =
453 :      1902      (MLER)<15,1>%,
454 :      M 1903      UNS =
455 :      1904      (MLER)<14,1>%,
456 :      M 1905      OPI =
457 :      1906      (MLER)<13,1>%,
458 :      M 1907      IAE =
459 :      1908      (MLER)<10,1>%,
460 :      M 1909      AOE =
461 :      1910      (MLER)<9,1>%,
462 :      M 1911      ECH_ERR =
463 :      1912      (MLER)<6,1>%,
464 :      M 1913      DPAR =
465 :      1914      (MLER)<5,1>%,
466 :      M 1915      CPAR =
467 :      1916      (MLER)<3,1>%,
468 :      M 1917      RMR =
469 :      1918      (MLER)<2,1>%,
470 :      M 1919      ILR =
471 :      1920      (MLER)<1,1>%,
472 :      M 1921      ILF =
473 :      1922      (MLER)<0,1>%,
474 :      M 1923      ARR_TYP =
475 :      1924      (MLMRT)<10,1>%,
476 :      M 1925      ML_NUM_ARR =
477 :      1926      (MLMRT)<11,5>%,
478 :      M 1927      REF_MAR =
479 :      1928      (MLMRT)<7,1>%,
480 :      M 1929      PROM_RW =
481 :      1930      (MLMR)>6,1>%,
482 :      M 1931      PROM_DIS =
483 :      1932      (MLMR)<5,1>%,
484 :      M 1933      DAT_CLK =
485 :      1934      (MLMRT)<4,1>%,
486 :      M 1935      DAT_DM =
487 :      1936      (MLMRT)<3,1>%,
488 :      M 1937      DCK_EN =
489 :      1938      (MLMR)>2,1>%,
490 :      M 1939      ECC_DIS =
```

DECLARATION SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

!MLDS BIT ASSIGNMENTS

!MLER BIT ASSIGNMENTS

!MLMR BIT ASSIGNMENTS

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

492 :ML4AD

DECLARATION SECTION

```
493 :  
494 :  
495 : *1940 (MLMR)<1,1>%,  
496 : M 1941 ECC_DM =  
497 : 1942 (MLMR)<0,1>%,  
498 : M 1943 DRV_TYP =  
499 : 1944 (MLDT)<0,1>%,  
500 : M 1945 CRC_A =  
501 : M 1946  
502 : 1947 (MLE1)<0,6>%,  
503 : M 1948 PAR_CRC_WRD =  
504 : 1949 (MLE1)>8,6>%,  
505 : M 1950 CRC_B =  
506 : M 1951  
507 : 1952 (MLE2)<0,6>%,  
508 : M 1953 UNC_ERR =  
509 : 1954 (MLEE)<15,1>%,  
510 : M 1955 SGL_ERR =  
511 : 1956 (MLEE)<14,1>%,  
512 : M 1957 CRC_ERR =  
513 : 1958 (MLEE)<13,1>%,  
514 : M 1959 BIT_IN_ERR =  
515 : 1960 (MLEE)>0,8>%,  
516 : M 1961 CHAN_IN_ERR =  
517 : 1962 (MLEE)<8,6>%,  
518 : 1963  
519 : 1964 : MISCELLANEOUS MACRO DEFINITIONS:  
520 : 1965  
521 : M 1966 SN3 =  
522 : 1967 (MLSN)<12,4>%,  
523 : M 1968 SN2 =  
524 : 1969 (MLSN)<8,4>%,  
525 : M 1970 SN1 =  
526 : 1971 (MLSN)<4,4>%,  
527 : M 1972 SNO =  
528 : 1973 (MLSN)<0,4>%,  
529 : M 1974 IS_SET =  
530 : M 1975  
531 : 1976 eql 1%,  
532 : M 1977 IS_NOT_SET =  
533 : M 1978  
534 : 1979 eql 0%,  
535 : M 1980 HI =  
536 : M 1981  
537 : 1982 ML_REG[.index,FORCE_HI]%,  
538 : M 1983 LO =  
539 : M 1984  
540 : 1985 ML_REG[.index,FORCE_LO]%,  
541 : M 1986 IGNORE =  
542 : M 1987  
543 : 1988 ML_REG[.index,DONT_CARE]%,  
544 : M 1989 MLE2_MASK =  
545 : M 1990  
546 : 1991 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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

```
548 :ML4AD
549 :
550 :
551 : M 1992
552 : M 1993
553 : M 1994
554 : M 1995
555 : M 1996
556 : M 1997
557 : M 1998
558 : M 1999
559 : M 2000
560 : M 2001
561 : M 2002
562 : M 2003
563 : M 2004
564 : M 2005
565 : M 2006
566 : M 2007
567 : M 2008
568 : M 2009
569 : M 2010
570 : M 2011
571 : M 2012
572 : M 2013
573 : M 2014
574 : M 2015
575 : M 2016
576 : M 2017
577 : M 2018
578 : M 2019
579 : M 2020
580 : M 2021
581 : M 2022
582 : M 2023
583 : M 2024
584 : M 2025
585 : M 2026
586 : M 2027
587 : M 2028
588 : M 2029
589 : M 2030
590 : M 2031
591 : M 2032
592 : M 2033
593 : M 2034
594 : M 2035
595 : M 2036
596 : M 2037
597 : M 2038
598 : M 2039
599 : M 2040
600 : M 2041
601 : M 2042
602 : M 2043

DECLARATION SECTION

WRT_MASK =
!GENERATE MASK DATA PATTERN USING REGISTER FORCE LO, HI AND IGNORE B
.IGNORE or ((not .LO) and (.HI or .TST_PAT))%,
CLR_MBUS =
!CLEAR MASS BUS RESTORE DRIVE NUMBER
CLR = ONE; DRV_NUM = .ML_DUT%,
<BLF/SYNONYM IS_SET = EQL 1 * >
<BLF/SYNONYM IS_NOT_SET = EQL 0 * >

DIAGNOSTIC DATA REGISTER MACROS

RD_LNG_WRD =
!READ DATA DIAG REGS INTO BIND LOCATIONS
D1_TEMP = .MLD1;
D2_TEMP = .MLD2;
E2_TEMP = .MLE2%,
WRT_LNG_WRD =
!LOADS DATA DIAG REG WITH CONTENTS OF BIND LOCATIONS
MLD1 = .D1_TEMP;
MLD2 = .D2_TEMP;
MLE2 = .E2_TEMP%,
TIME_OUT_LOOP =
!WAIT LOOP DURING MASS BUS TRANSFER
do
0
until .DRY:%,

ERROR PRINTING THRESHOLD MACROS

CLR_THRESHOLD =
P_CNT = ZERO:%,
CMP_THRESHOLD =
P_CNT = .P_CNT + 1;
if .P_CNT gtr .LIMIT
then exitloop:%;

BIND DECLARATIONS

bind
!MANUAL INTERVENTION TEST ERROR AND OPERATOR PROMPT MESSAGES.
!TESTS 21 'POWER FAIL TEST' AND TEST 61 'BATTERY BACK-UP TEST'
T_21 = uplit (%asciz'***** MANUAL INTERVENTION TEST 21 POWER FAIL *****'),
T_61 = uplit (%asciz'***** MANUAL INTERVENTION TEST 61 BATTERY BACK-UP *****'),
PDR_OFF = uplit (%asciz'WITH BATTERY BACK-UP ON TURN BOX AC CIRCUIT BREAKER OFF'),
PWR_ON = uplit (%asciz'WITH BATTERY BACK-UP OFF TURN BOX AC CIRCUIT BREAKER ON'),
UNS_ERR = uplit (%asciz'UNS BIT NOT SET DURING AC POWER FAIL'),
VV_NOT_SET = uplit (%asciz'VV BIT NOT SET AFTER RD_IN_PRESET FUNCTION'),
```

604 :ML4AD

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

DECLARATION SECTION

607 : 2044 VV_CLEAR = uplit (%asciz'VV BIT CLEARED AFTER RD-IN-PRESET FUNCTION'),
608 : 2045 NO_INIT = uplit (%asciz'MEMORY NOT INITIALIZED AFTER POWER UP WITH BAD BATTERY BACK-UP'),
609 : 2046 MB_DIS_ERR = uplit (%asciz'REGISTERS NOT CLEARED BY MB DIS AFTER POWER FAIL'),
610 : 2047 WC_ERR = uplit (%asciz'SC BIT SET DURING WRITE CHECK TRANSFER'),
611 : 2048 W_ERR = uplit (%asciz'SC BIT SET DURING WRITE TRANSFER'),
612 : 2049 VV_SET = uplit (%asciz'VV BIT STILL SET AFTER POWER LOSS WITH BAD BATTERY BACK-UP'),
613 : 2050 PUP_BB = uplit (%asciz'WITH BATTERY BACK-UP ON TURN BOX AC CIRCUIT BREAKER ON'),
614 : 2051 BB_VV_ERR = uplit (%asciz'VV BIT CLEARED AFTER POWER FAIL WITH GOOD BATTERY BACK-UP'),
615 : 2052 BB_INIT_ERR = uplit (%asciz'MEMORY INITED AFTER POWER FAIL WITH GOOD BATTERY BACK-UP'),
616 : 2053 BB_BB_ERR = uplit (%asciz'BATTERY BACK-UP FAILED TO MAINTAIN MEMORY INTEGRITY DURING POWER FAIL'),
617 : 2054

ERROR DATA MAPPING FORMATS

618 : 2055
619 : 2056
620 : 2057 FMT_1 = uplit (%asciz'%AEXPECTED: %06XA READ: %06XNZN'),
621 : 2058 FMT_2 = uplit (%asciz'%AGOOD DATA: %06XA BAD DATA: %06XA XOR: %06XNZN'),
622 : 2059 FMT_3 = uplit (%asciz'%ADRIE SN: %06XNZN'),
623 : 2060 FMT_4 = uplit (%asciz'%ABIT IN ERROR: %06XNZN'),
624 : 2061 FMT_5 = uplit (%asciz'%AGOOD NIB DATA: %02XA BAD NIB DATA: %02XA NIB POS: %04XNZN'),
625 : 2062 FMT_6 = uplit (%asciz'%ANIB IN ERROR: %D4XNZN'),
626 : 2063 FMT_7 = uplit (%asciz'%AFAILED AT: %06XNZN'),
627 : 2064 FMT_8 = uplit (%asciz'%AREPLACE ARR MOD: %D2XNZN'),
628 : 2065 FMT_9 = uplit (%asciz'%AFAILED AT DSA: %06XNZN'),
629 : 2066 FMT_10 = uplit (%asciz'%ABIT<15:10>: %B6XA BIT<9:0>: %B10XNZN'),
630 : 2067 FMT_11 = uplit (%asciz'%AFAILING REG ADRS: %06XNZN'),
631 : 2068 FMT_12 = uplit (%asciz'%AFAILING FUNC: %06XNZN'),
632 : 2069 FMT_13 = uplit (%asciz'%AOFF SET CNT FOR NIB : %D2 %A = %D2 XNZN'),
633 : 2070 FMT_14 = uplit (%asciz'%AWROTE: %D2XA READ: %D2XNZN'),
634 : 2071 FMT_15 = uplit (%asciz'%ANIBBLES XFERED BEFORE ERROR: %D5ZN'),
635 : 2072 FMT_16 = uplit (%asciz'%AFAILING REG: %06XA GOOD DATA: %06XA BAD DATA: %06XNZN'),
636 : 2073 FMT_17 = uplit (%asciz'%NZADIAGNOSING UNIT %01XNZN'),
637 : 2074 FMT_18 = uplit (%asciz'%ATIMED OUT DURING MBUS %02XA FUNCXNZN'),
638 : 2075 FMT_19 = uplit (%asciz'%ACRC GEN = B: %B6XA A: %B6XA P: %B6XNZN'),
639 : 2076 FMT_20 = uplit (%asciz'%ACRC CAL = B: %B6XA A: %B6XA P: %B6XNZN'),
640 : 2077 FMT_21 = uplit (%asciz'%AFAILED AT PLOG: %D2XA CHANNEL: %D2XNZN'),
641 : 2078 FMT_22 = uplit (%asciz'%AFAILED AT WRD: %D2XA BIT: %D2XNZN'),
642 : 2079 FMT_23 = uplit (%asciz'%AREGISTER%3%AADDRESS%5%AACONTENTS%7XNZN'),
643 : 2080 FMT_24 = uplit (%asciz'%S%T%5%06%S4%016XN'),
644 : 2081 FMT_25 = uplit (%asciz'%S4%TZA: %S%06XN'),
645 : 2082 FMT_26 = uplit (%asciz'%S4%TZA: %S%D1%D1%D1%D1XN'),
646 : 2083

ERROR MESSAGE MAPPING FORMATS

647 : 2084
648 : 2085
649 : 2086 ONE_FMT = uplit (%asciz'%TZN'),
650 : 2087 TWO_FMT = uplit (%asciz'%T%TZN'),
651 : 2088 THR_FMT = uplit (%asciz'%T%T%TZN'),
652 : 2089 FOR_FMT = uplit (%asciz'%T%T%T%TZN'),
653 : 2090 FIV_FMT = uplit (%asciz'%T%T%T%T%TZN'),
654 : 2091 SIX_FMT = uplit (%asciz'%T%T%T%T%T%TZN'),
655 : 2092 SEV_FMT = uplit (%asciz'%T%T%T%T%T%T%TZN'),
656 : 2093 EIG_FMT = uplit (%asciz'%T%T%T%T%T%T%T%TZN'),
657 : 2094 NIN_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%TZN'),
658 : 2095 TEN_FMT = uplit (%asciz'%T%T%T%T%T%T%T%T%T%TZN'),

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

660 :ML4AD
661 :
662 :
663 :
664 :
665 :
666 :
667 :
668 :
669 :
670 :
671 :
672 :
673 :
674 :
675 :
676 :
677 :
678 :
679 :
680 :
681 :
682 :
683 :
684 :
685 :
686 :
687 :
688 :
689 :
690 :
691 :
692 :
693 :
694 :
695 :
696 :
697 :
698 :
699 :
700 :
701 :
702 :
703 :
704 :
705 :
706 :
707 :
708 :
709 :
710 :
711 :
712 :
713 :
714 :

DECLARATION SECTION

ELV_FMT = uplit (%asciz'XTXTXTXTXTXTXTXTXTXTXTN'),

DIAGNOSTIC VOCABULARY

WORDS

WRD_1 = uplit (%asciz' GO'),
WRD_2 = uplit (%asciz' DRV_RDY'),
WRD_3 = uplit (%asciz' ILF'),
WRD_4 = uplit (%asciz' OPI'),
WRD_5 = uplit (%asciz' BAD'),
WRD_6 = uplit (%asciz' GOOD'),
WRD_7 = uplit (%asciz' PARITY_NOT'),
WRD_8 = uplit (%asciz' GENERATED'),
WRD_9 = uplit (%asciz' DETECTED'),
WRD_10 = uplit (%asciz' ERROR'),
WRD_11 = uplit (%asciz' AFTER'),
WRD_12 = uplit (%asciz' DURING'),
WRD_13 = uplit (%asciz' AT'),
WRD_14 = uplit (%asciz' FAILURE'),
WRD_15 = uplit (%asciz' ATA'),
WRD_16 = uplit (%asciz' ATTN'),
WRD_17 = uplit (%asciz' WRITING'),
WRD_18 = uplit (%asciz' VV'),
WRD_19 = uplit (%asciz' FUNC'),
WRD_20 = uplit (%asciz' TRE'),
WRD_21 = uplit (%asciz' RMR'),
WRD_22 = uplit (%asciz' EXCESSIVE'),
WRD_23 = uplit (%asciz' MBUS'),
WRD_24 = uplit (%asciz' DATA'),
WRD_25 = uplit (%asciz' CONTINUITY'),
WRD_26 = uplit (%asciz' AOE'),
WRD_27 = uplit (%asciz' LBT'),
WRD_29 = uplit (%asciz' PREMATURELY'),
WRD_30 = uplit (%asciz' IAE'),
WRD_31 = uplit (%asciz' INCREMENT'),
WRD_32 = uplit (%asciz' WITH'),
WRD_33 = uplit (%asciz' UV'),
WRD_34 = uplit (%asciz' UNS'),
WRD_35 = uplit (%asciz' PROM'),
WRD_36 = uplit (%asciz' OR'),
WRD_37 = uplit (%asciz' SELECT'),
WRD_38 = uplit (%asciz' REG'),
WRD_39 = uplit (%asciz' UNIQUE'),
WRD_40 = uplit (%asciz' 14'),
WRD_41 = uplit (%asciz' NIBBLE_CNT'),
WRD_42 = uplit (%asciz' GTR'),
WRD_43 = uplit (%asciz' WHILE'),
WRD_44 = uplit (%asciz' TRE'),
WRD_45 = uplit (%asciz' INITIAL'),
WRD_46 = uplit (%asciz' OFF_SET'),

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

716 ;ML4AD
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 :
758 :
759 :
760 :
761 :
762 :
763 :
764 :
765 :
766 :
767 :
768 :
769 :
770 :

DECLARATION SECTION

```

WRD_47 = uplit (%asciz' COUNT'),
WRD_48 = uplit (%asciz' DELAY'),
WRD_49 = uplit (%asciz' TESTS'),
WRD_50 = uplit (%asciz' ADRS'),
WRD_51 = uplit (%asciz' COUNTER'),
WRD_52 = uplit (%asciz' REG'),
WRD_53 = uplit (%asciz' TESTED'),
WRD_54 = uplit (%asciz' NIBBLE'),
WRD_55 = uplit (%asciz' ALL'),
WRD_56 = uplit (%asciz' TEST'),
WRD_57 = uplit (%asciz' XFERED'),
WRD_58 = uplit (%asciz' NIBBLES'),
WRD_59 = uplit (%asciz' SC'),
WRD_60 = uplit (%asciz' MULTIPLEXER'),
WRD_61 = uplit (%asciz' UNEXPECTED'),
WRD_62 = uplit (%asciz' NED'),
WRD_63 = uplit (%asciz' ILR'),
WRD_64 = uplit (%asciz' CRC'),
WRD_65 = uplit (%asciz' SGL'),
WRD_67 = uplit (%asciz' ECH'),
WRD_68 = uplit (%asciz' UNC'),
WRD_69 = uplit (%asciz' BIT'),
WRD_70 = uplit (%asciz' CHANNEL'),
WRD_71 = uplit (%asciz' LATCH'),
WRD_72 = uplit (%asciz' DCK'),
WRD_73 = uplit (%asciz' BUS'),
WRD_74 = uplit (%asciz' SYNDROME'),
WRD_75 = uplit (%asciz' DECODE'),
WRD_76 = uplit (%asciz' NOT'),
WRD_77 = uplit (%asciz' UNCORRECTABLE'),
    
```

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'),
    
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

772 :ML4AD
773 :
774 :
775 :
776 :
777 :
778 :
779 :
780 :
781 :
782 :
783 :
784 :
785 :
786 :
787 :
788 :
789 :
790 :
791 :
792 :
793 :
794 :
795 :
796 :
797 :
798 :
799 :
800 :
801 :
802 :
803 :
804 :
805 :
806 :
807 :
808 :
809 :
810 :
811 :
812 :
813 :
814 :
815 :
816 :
817 :
818 :
819 :
820 :
821 :
822 :
823 :
824 :
825 :
826 :

DECLARATION SECTION

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' M_CS1'),
REG_2 = uplit (%asciz' MLDS'),
REG_3 = uplit (%asciz' MLER'),
REG_4 = uplit (%asciz' MLMR'),
REG_5 = uplit (%asciz' MLAS'),
REG_6 = uplit (%asciz' MLDA'),
REG_7 = uplit (%asciz' MLDT'),
REG_8 = uplit (%asciz' MLPA'),
REG_9 = uplit (%asciz' MLSN'),
REG_10 = uplit (%asciz' MLE1'),
REG_11 = uplit (%asciz' MLE2'),
REG_12 = uplit (%asciz' MLD1'),
REG_13 = uplit (%asciz' MLD2'),
REG_14 = uplit (%asciz' MLEE'),
REG_15 = uplit (%asciz' MLEL'),
REG_16 = uplit (%asciz' MLPD'),
REG_17 = uplit (%asciz' MLCS2'),
REG_18 = uplit (%asciz' MLWC'),
REG_19 = uplit (%asciz' MLBA'),
REG_20 = uplit (%asciz' MLBA1'),
REG_21 = uplit (%asciz' MLCS3').

MODULES IN ERROR MESSAGES

ASYNCR = uplit (%asciz' ASYNCHRONOUS FAILURE MODULE 7361'),
SYNCR = uplit (%asciz' SYNCHRONOUS FAILURE MODULE 7362'),
ARR_DAT = uplit (%asciz' ARRAY DATA FAILURE MODULE 7363'),
MEM_ARR = uplit (%asciz' MEMORY ARRAY FAILURE MODULE 7357').

828 :ML4AD
829 :
830 :
831 :
832 :
833 :
834 :
835 :
836 :
837 :
838 :
839 :
840 :
841 :
842 :
843 :
844 :
845 :

DECLARATION SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (3)

2252 INTER = uplit (%asciz'INTERMEDIATE DIAGNOSTIC MESSAGE'),
2253 DATA_LATE = uplit (%asciz'DATA LATE ERROR DURING TRANSFER'),
2254 SC_SET = uplit (%asciz'SC BIT SET DURING TRANSFER'),
2255 TRBLE_LOOP = uplit (%asciz'TROUBLE SHOOT LOOP ERRORS'),
2256 RH_ERROR = uplit (%asciz'RH CONTROLLER ERRORS'),
2257 TIME_OUT = uplit (%asciz'DRIVE HUNG AFTER MASS BUS TRANSFER'),

DATA DIAGNOSTIC REGISTER SAVE LOCATIONS

2261 D1_TEMP = NIB_SAVE,
2262 D2_TEMP = NIB_SAVE [1, 0, 16, 0];
2263 E2_TEMP = NIB_SAVE [2, 0, 16, 0];

2265 %SBTTL 'ROUTINE DECLARATION SECTION'
2266

847 :ML4AD
848 :
849 :
850 :
851 :
852 :
853 :
854 :
855 :
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 :

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (4)

2267 routine LOAD_STACK (STK_PTR, NIB_PTR) : novalue =
2268 begin

2269
2270 !++
2271 FUNCTIONAL DESCRIPTION:
2272 LOAD STACK TAKES GOOD NIBBLE DATA
2273 FOUND IN THE STRUCTURE 'NIB SAVE'
2274 AND STORES IT INTO THE STRUCTURE
2275 'STACK' REWRITING ANY BAD NIBBLE
2276 'STACK' LOCATIONS WITH GOOD NIBBLE
2277 DATA

2278
2279 FORMAL PARAMETERS:
2280 STK_PTR
2281 POINTS TO PRESENT DEPTH OF THE
2282 'STACK' WHERE PRESENT GOOD NIBBLE
2283 DATA IS TO BE STORED.

2284
2285 NIB_PTR
2286 POINTS TO CURRENT NIBBLE POSITION BEING
2287 MANIPULATED.

2288
2289 IMPLICIT INPUTS:
2290 STACK
2291 VECTOR OF 198 BYTE LOCATIONS WHERE
2292 GOOD NIBBLE DATA IS STORED
2293 DURING DIAGNOSTIC MODE READS, AFTER
2294 BAD NIBBLE LOCATIONS HAVE BEEN
2295 STRIPPED AWAY.

2296
2297 STK_OFF
2298
2299 vector of 9 byte LOCATIONS WHICH
2300 STORES AWAY A BAD NIBBLE OFF SET
2301 COUNT FOR EACH NIBBLE POSITION

2302
2303 NIB_SAVE
2304 BLOCK OF 3 WORDS TO STORE THE
2305 DATA FOUND IN MLD1, MLD2 AND
2306 MLE2 AFTER A DIAGNOSTIC MODE READ.

2307
2308 IMPLICIT OUTPUTS:
2309 'STACK' LOADED WITH GOOD NIBBLE
2310 DATA

2311
2312 COMPLETETION CODES: NONE

2313
2314 SIDE EFFECTS: NONE

2315
2316 --

2317
2318 case .NIB_PTR from 0 to 9 of

!SELECT NIBBLE DATA TO BE LOADED INTO THE STACK

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (4)

903 :ML4AD
904 :
905 :
906 :
907 :
908 :
909 :
910 :
911 :
912 :
913 :
914 :
915 :
916 :
917 :
918 :
919 :
920 :
921 :
922 :
923 :
924 :
925 :
926 :
927 :
928 :
929 :
930 :
931 :
932 :
933 :
934 :
935 :
936 :
937 :
938 :
939 :
940 :
941 :
942 :
943 :
944 :
945 :
946 :
947 :
948 :
949 :
953 :
954 :
955 :

ROUTINE DECLARATION SECTION

```
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;
```


ROUTINE DECLARATION SECTION

```
957      :ML4AD  
958      :  
959      :  
960 004716 052 052 052 P.AAA: .ASCII /***/  
961 004721 052 052 052 .ASCII /***/  
962 004724 052 052 052 .ASCII /***/  
963 004727 040 115 101 .ASCII / MA/  
964 004732 116 125 101 .ASCII /NUA/  
965 004735 114 040 111 .ASCII /L I/  
966 004740 116 124 105 .ASCII /NTE/  
967 004743 122 126 105 .ASCII /RVE/  
968 004746 116 124 111 .ASCII /NTI/  
969 004751 117 116 040 .ASCII /ON /  
970 004754 124 105 123 .ASCII /TES/  
971 004757 124 040 062 .ASCII /T 2/  
972 004762 061 040 120 .ASCII /I P/  
973 004765 117 127 105 .ASCII /OWE/  
974 004770 122 040 106 .ASCII /R F/  
975 004773 101 111 114 .ASCII /AIL/  
976 004776 040 052 052 .ASCII / **/  
977 005001 052 052 052 .ASCII /***/  
978 005004 052 052 052 .ASCII /***/  
979 005007 052 000 000 .ASCII /*/<00><00>  
980 005012 052 052 052 P.AAB: .ASCII /***/  
981 005015 052 052 052 .ASCII /***/  
982 005020 052 052 052 .ASCII /***/  
983 005023 040 115 101 .ASCII / MA/  
984 005026 116 125 101 .ASCII /NUA/  
985 005031 114 040 111 .ASCII /L I/  
986 005034 116 124 105 .ASCII /NTE/  
987 005037 122 126 105 .ASCII /RVE/  
988 005042 116 124 111 .ASCII /NTI/  
989 005045 117 116 040 .ASCII /ON /  
990 005050 124 105 123 .ASCII /TES/  
991 005053 124 040 066 .ASCII /T 6/  
992 005056 061 040 102 .ASCII /I B/  
993 005061 101 124 124 .ASCII /ATT/  
994 005064 105 122 131 .ASCII /ERY/  
995 005067 040 102 101 .ASCII / BA/  
996 005072 103 113 055 .ASCII /CK-/  
997 005075 125 120 040 .ASCII /UP /  
998 005100 052 052 052 .ASCII /***/  
999 005103 052 052 052 .ASCII /***/  
1000 005106 052 052 052 .ASCII /***/  
1001 005111 000 .ASCII <00>  
1002 005112 127 111 124 P.AAC: .ASCII /WIT/  
1003 005115 110 040 102 .ASCII /H B/  
1004 005120 101 124 124 .ASCII /ATT/  
1005 005123 105 122 131 .ASCII /ERY/  
1006 005126 040 102 101 .ASCII / BA/  
1007 005131 103 113 055 .ASCII /CK-/  
1008 005134 125 120 040 .ASCII /UP /  
1009 005137 117 116 040 .ASCII /ON /  
1010 005142 124 125 122 .ASCII /TUR/  
1011 005145 116 040 102 .ASCII /N B/
```

1013										
1014										
1015										
1016	005150	117	130	040						
1017	005153	101	103	040						
1018	005156	103	111	122						
1019	005161	103	125	111						
1020	005164	124	040	102						
1021	005167	122	105	101						
1022	005172	113	105	122						
1023	005175	040	117	106						
1024	005200	106	000							
1025	005202	127	111	124	P.AAD:					
1026	005205	110	040	102						
1027	005210	101	124	124						
1028	005213	105	122	131						
1029	005216	040	102	101						
1030	005221	103	113	055						
1031	005224	125	120	040						
1032	005227	117	106	106						
1033	005232	040	124	125						
1034	005235	122	116	040						
1035	005240	102	117	130						
1036	005243	040	101	103						
1037	005246	040	103	111						
1038	005251	122	103	125						
1039	005254	111	124	040						
1040	005257	102	122	105						
1041	005262	101	113	105						
1042	005265	122	040	117						
1043	005270	116	000							
1044	005272	125	116	123	P.AAE:					
1045	005275	040	102	111						
1046	005300	124	040	116						
1047	005303	117	124	040						
1048	005306	123	105	124						
1049	005311	040	104	125						
1050	005314	122	111	116						
1051	005317	107	040	101						
1052	005322	103	040	120						
1053	005325	117	127	105						
1054	005330	122	040	106						
1055	005333	101	111	114						
1056	005336	000	000							
1057	005340	126	126	040	P.AAF:					
1058	005343	102	111	124						
1059	005346	040	116	117						
1060	005351	124	040	123						
1061	005354	105	124	040						
1062	005357	101	106	124						
1063	005362	105	122	040						
1064	005365	122	104	137						
1065	005370	111	116	137						
1066	005373	120	122	105						
1067	005376	123	105	124						

```

:ML4AD
:
ROUTINE DECLARATION SECTION
.ASCII /OX /
.ASCII /AC /
.ASCII /CIR/
.ASCII /CUI/
.ASCII /T B/
.ASCII /REA/
.ASCII /KER/
.ASCII / OF/
.ASCII /F/<00>
P.AAD:.ASCII /WIT/
.ASCII /H B/
.ASCII /ATT/
.ASCII /ERY/
.ASCII / BA/
.ASCII /CK-/
.ASCII /UP /
.ASCII /OFF/
.ASCII / TU/
.ASCII /RN /
.ASCII /BOX/
.ASCII / AC/
.ASCII / CI/
.ASCII /RCU/
.ASCII /IT /
.ASCII /BRE/
.ASCII /AKE/
.ASCII /R O/
.ASCII /N/<00>
P.AAE:.ASCII /UNS/
.ASCII / BI/
.ASCII /T N/
.ASCII /OT /
.ASCII /SET/
.ASCII / DU/
.ASCII /RIN/
.ASCII /G A/
.ASCII /C P/
.ASCII /OWE/
.ASCII /R F/
.ASCII /AIL/
P.AAF:.ASCII <00><00>
.ASCII /VV /
.ASCII /BIT/
.ASCII / NO/
.ASCII /T S/
.ASCII /ET /
.ASCII /AFT/
.ASCII /ER /
.ASCII /RD /
.ASCII /IN~/
.ASCII /PRE/
.ASCII /SET/

```

Line No.	Code	Value 1	Value 2	Value 3	Value 4	Label	Routine	Declaration
1069						:ML4AD		
1070						:		ROUTINE DECLARATION SECTION
1071								
1072	005401	040	106	125			.ASCII	/FU/
1073	005404	116	103	124			.ASCII	/NCT/
1074	005407	111	117	116			.ASCII	/ION/
1075	005412	000	000				.ASCII	<00><00>
1076	005414	126	126	040	P.AAG:		.ASCII	/VV /
1077	005417	102	111	124			.ASCII	/BIT/
1078	005422	040	103	114			.ASCII	/CL/
1079	005425	105	101	122			.ASCII	/EAR/
1080	005430	105	104	040			.ASCII	/ED /
1081	005433	101	106	124			.ASCII	/AFT/
1082	005436	105	122	040			.ASCII	/ER /
1083	005441	122	104	055			.ASCII	/RD-/
1084	005444	111	116	055			.ASCII	/IN-/
1085	005447	120	122	105			.ASCII	/PRE/
1086	005452	123	105	124			.ASCII	/SET/
1087	005455	040	106	125			.ASCII	/FU/
1088	005460	116	103	124			.ASCII	/NCT/
1089	005463	111	117	116			.ASCII	/ION/
1090	005466	000	000				.ASCII	<00><00>
1091	005470	115	105	115	P.AAH:		.ASCII	/MEM/
1092	005473	117	122	131			.ASCII	/ORY/
1093	005476	040	116	117			.ASCII	/NO/
1094	005501	124	040	111			.ASCII	/T I/
1095	005504	116	111	124			.ASCII	/NIT/
1096	005507	111	101	114			.ASCII	/IAL/
1097	005512	111	132	105			.ASCII	/IZE/
1098	005515	104	040	101			.ASCII	/D A/
1099	005520	106	124	105			.ASCII	/FTE/
1100	005523	122	040	120			.ASCII	/R P/
1101	005526	117	127	105			.ASCII	/OWE/
1102	005531	122	040	125			.ASCII	/R U/
1103	005534	120	040	127			.ASCII	/P W/
1104	005537	111	124	110			.ASCII	/ITH/
1105	005542	040	102	101			.ASCII	/BA/
1106	005545	104	040	102			.ASCII	/D B/
1107	005550	101	124	124			.ASCII	/ATT/
1108	005553	105	122	131			.ASCII	/ERY/
1109	005556	040	102	101			.ASCII	/BA/
1110	005561	103	113	055			.ASCII	/CK-/
1111	005564	125	120	000			.ASCII	/UP/<00>
1112	005567	000					.ASCII	<00>
1113	005570	122	105	107	P.AAI:		.ASCII	/REG/
1114	005573	111	123	124			.ASCII	/IST/
1115	005576	105	122	123			.ASCII	/ERS/
1116	005601	040	116	117			.ASCII	/NO/
1117	005604	124	040	103			.ASCII	/T C/
1118	005607	114	105	101			.ASCII	/LEA/
1119	005612	122	105	104			.ASCII	/RED/
1120	005615	040	102	131			.ASCII	/BY/
1121	005620	040	115	102			.ASCII	/MB/
1122	005623	137	104	111			.ASCII	/DI/
1123	005626	123	040	101			.ASCII	/S A/

```

1125      :ML4AD
1126      :
1127      :
1128 005631 106 124 105 .ASCII /FTE/
1129 005634 122 040 120 .ASCII /R P/
1130 005637 117 127 105 .ASCII /OWE/
1131 005642 122 040 106 .ASCII /R F/
1132 005645 101 111 114 .ASCII /AIL/
1133 005650 000 000 .ASCII <00><00>
1134 005652 123 103 040 P.AAJ: .ASCII /SC /
1135 005655 102 111 124 .ASCII /BIT/
1136 005660 040 123 105 .ASCII / SE/
1137 005663 124 040 104 .ASCII /T D/
1138 005666 125 122 111 .ASCII /URI/
1139 005671 116 107 040 .ASCII /NG /
1140 005674 127 122 111 .ASCII /WRI/
1141 005677 124 105 040 .ASCII /TE /
1142 005702 103 110 105 .ASCII /CHE/
1143 005705 103 113 040 .ASCII /CK /
1144 005710 124 122 101 .ASCII /TRA/
1145 005713 116 123 106 .ASCII /NSF/
1146 005716 105 122 000 .ASCII /ER/<00>
1147 005721 000 .ASCII <00>
1148 005722 123 103 040 P.AAK: .ASCII /SC /
1149 005725 102 111 124 .ASCII /BIT/
1150 005730 040 123 105 .ASCII / SE/
1151 005733 124 040 104 .ASCII /T D/
1152 005736 125 122 111 .ASCII /URI/
1153 005741 116 107 040 .ASCII /NG /
1154 005744 127 122 111 .ASCII /WRI/
1155 005747 124 105 040 .ASCII /TE /
1156 005752 124 122 101 .ASCII /TRA/
1157 005755 116 123 106 .ASCII /NSF/
1158 005760 105 122 000 .ASCII /ER/<00>
1159 005763 000 .ASCII <00>
1160 005764 126 126 040 P.AAL: .ASCII /VV /
1161 005767 102 111 124 .ASCII /BIT/
1162 005772 040 123 124 .ASCII / ST/
1163 005775 111 114 114 .ASCII /ILL/
1164 006000 040 123 105 .ASCII / SE/
1165 006003 124 040 101 .ASCII /T A/
1166 006006 106 124 105 .ASCII /FTE/
1167 006011 122 040 120 .ASCII /R P/
1168 006014 117 127 105 .ASCII /OWE/
1169 006017 122 040 114 .ASCII /R L/
1170 006022 117 123 123 .ASCII /OSS/
1171 006025 040 127 111 .ASCII / WI/
1172 006030 124 110 040 .ASCII /TH /
1173 006033 102 101 104 .ASCII /BAD/
1174 006036 040 102 101 .ASCII / BA/
1175 006041 124 124 105 .ASCII /TTE/
1176 006044 122 131 040 .ASCII /RY /
1177 006047 102 101 103 .ASCII /BAC/
1178 006052 113 055 125 .ASCII /K-U/
1179 006055 120 000 000 .ASCII /P/<00><00>

```

1181					:ML4AD				
1182					:				ROUTINE DECLARATION SECTION
1183					:				
1184	006060	127	111	124	P.AAM:	.ASCII	/WIT/		
1185	006063	110	040	102		.ASCII	/H B/		
1186	006066	101	124	124		.ASCII	/ATT/		
1187	006071	105	122	131		.ASCII	/ERY/		
1188	006074	040	102	101		.ASCII	/ BA/		
1189	006077	103	113	055		.ASCII	/CK-/		
1190	006102	125	120	040		.ASCII	/UP /		
1191	006105	117	116	040		.ASCII	/ON /		
1192	006110	124	125	122		.ASCII	/TUR/		
1193	006113	116	040	102		.ASCII	/N B/		
1194	006116	117	130	040		.ASCII	/OX /		
1195	006121	101	103	040		.ASCII	/AC /		
1196	006124	103	111	122		.ASCII	/CIR/		
1197	006127	103	125	111		.ASCII	/CUI/		
1198	006132	124	040	102		.ASCII	/T B/		
1199	006135	122	105	101		.ASCII	/REA/		
1200	006140	113	105	122		.ASCII	/KER/		
1201	006143	040	117	116		.ASCII	/ ON/		
1202	006146	000	000			.ASCII	<00><00>		
1203	006150	126	126	040	P.AAN:	.ASCII	/VV /		
1204	006153	102	111	124		.ASCII	/BIT/		
1205	006156	040	103	114		.ASCII	/ CL/		
1206	006161	105	101	122		.ASCII	/EAR/		
1207	006164	105	104	040		.ASCII	/ED /		
1208	006167	101	106	124		.ASCII	/AFT/		
1209	006172	105	122	040		.ASCII	/ER /		
1210	006175	120	117	127		.ASCII	/POW/		
1211	006200	105	122	040		.ASCII	/ER /		
1212	006203	106	101	111		.ASCII	/FAI/		
1213	006206	114	040	127		.ASCII	/L W/		
1214	006211	111	124	110		.ASCII	/ITH/		
1215	006214	040	107	117		.ASCII	/ GO/		
1216	006217	117	104	040		.ASCII	/OD /		
1217	006222	102	101	124		.ASCII	/BAT/		
1218	006225	124	105	122		.ASCII	/TER/		
1219	006230	131	040	102		.ASCII	/Y B/		
1220	006233	101	103	113		.ASCII	/ACK/		
1221	006236	055	125	120		.ASCII	/-UP/		
1222	006241	000				.ASCII	<00>		
1223	006242	115	105	115	P.AAO:	.ASCII	/MEM/		
1224	006245	117	122	131		.ASCII	/ORY/		
1225	006250	040	111	116		.ASCII	/ IN/		
1226	006253	111	124	105		.ASCII	/ITE/		
1227	006256	104	040	101		.ASCII	/D A/		
1228	006261	106	124	105		.ASCII	/FTE/		
1229	006264	122	040	120		.ASCII	/R P/		
1230	006267	117	127	105		.ASCII	/OWE/		
1231	006272	122	040	106		.ASCII	/R F/		
1232	006275	101	111	114		.ASCII	/AIL/		
1233	006300	040	127	111		.ASCII	/ WI/		
1234	006303	124	110	040		.ASCII	/TH /		
1235	006306	107	117	117		.ASCII	/GOO/		

1237									
1238									
1239									
1240	006311	1J4	040	102				.ASCII	/D B/
1241	006314	101	124	124				.ASCII	/ATT/
1242	006317	105	122	131				.ASCII	/ERY/
1243	006322	040	102	101				.ASCII	/ BA/
1244	006325	103	113	055				.ASCII	/CK-/
1245	006330	125	120	000				.ASCII	/UP/<00>
1246	006333	000						.ASCII	<00>
1247	006334	102	101	124	P.AAP:			.ASCII	/BAT/
1248	006337	124	105	122				.ASCII	/TER/
1249	006342	131	040	102				.ASCII	/Y B/
1250	006345	101	103	113				.ASCII	/ACK/
1251	006350	055	125	120				.ASCII	/-UP/
1252	006353	040	106	101				.ASCII	/ FA/
1253	006356	111	114	105				.ASCII	/ILE/
1254	006361	104	040	124				.ASCII	/D T/
1255	006364	117	040	115				.ASCII	/O M/
1256	006367	101	111	116				.ASCII	/AIN/
1257	006372	124	101	111				.ASCII	/TAI/
1258	006375	116	040	115				.ASCII	/N M/
1259	006400	105	115	117				.ASCII	/EMO/
1260	006403	122	131	040				.ASCII	/RY /
1261	006406	111	116	124				.ASCII	/INT/
1262	006411	105	107	122				.ASCII	/EGR/
1263	006414	111	124	131				.ASCII	/ITY/
1264	006417	040	104	125				.ASCII	/ DU/
1265	006422	122	111	116				.ASCII	/RIN/
1266	006425	107	040	120				.ASCII	/G P/
1267	006430	117	127	105				.ASCII	/OWE/
1268	006433	122	040	106				.ASCII	/R F/
1269	006436	101	111	114				.ASCII	/AIL/
1270	006441	000						.ASCII	<00>
1271	006442	045	101	105	P.AAQ:			.ASCII	/XAE/
1272	006445	130	120	105				.ASCII	/XPE/
1273	006450	103	124	105				.ASCII	/CTE/
1274	006453	104	072	040				.ASCII	/D: /
1275	006456	045	117	066				.ASCII	/X06/
1276	006461	045	101	040				.ASCII	/XA /
1277	006464	040	040	040				.ASCII	/ /
1278	006467	122	105	101				.ASCII	/REA/
1279	006472	104	072	040				.ASCII	/D: /
1280	006475	045	117	066				.ASCII	/X06/
1281	006500	045	116	045				.ASCII	/XNZ/
1282	006503	116	000	000				.ASCII	/N/<00><00>
1283	006506	045	101	107	P.AAR:			.ASCII	/XAG/
1284	006511	117	117	104				.ASCII	/OOD/
1285	006514	040	104	101				.ASCII	/ DA/
1286	006517	124	101	072				.ASCII	/TA:/
1287	006522	040	045	117				.ASCII	/ X0/
1288	006525	066	045	101				.ASCII	/6XA/
1289	006530	040	040	040				.ASCII	/ /
1290	006533	040	102	101				.ASCII	/ BA/
1291	006536	104	040	104				.ASCII	/D D/


```

1293          :ML4AD
1294          :
1295          :
1296 006541    101    124    101    .ASCII /ATA/
1297 006544    072    040    045    .ASCII /: %/
1298 006547    117    066    045    .ASCII /06%/
1299 006552    101    040    040    .ASCII /A /
1300 006555    040    040    130    .ASCII / X/
1301 006560    117    122    072    .ASCII /OR:/
1302 006563    040    045    117    .ASCII / %0/
1303 006566    066    045    116    .ASCII /6%N/
1304 006571    045    116    000    .ASCII /%N/<00>
1305 006574    045    101    104    P.AAS: .ASCII /%AD/
1306 006577    122    111    126    .ASCII /RIV/
1307 006602    105    040    123    .ASCII /E S/
1308 006605    116    072    040    .ASCII /N: /
1309 006610    045    117    066    .ASCII /%06/
1310 006613    045    116    045    .ASCII /%N%/
1311 006616    116    000    .ASCII /N/<00>
1312 006620    045    101    102    P.AAT: .ASCII /%AB/
1313 006623    111    124    040    .ASCII /IT /
1314 006626    111    116    040    .ASCII /IN /
1315 006631    105    122    122    .ASCII /ERR/
1316 006634    117    122    072    .ASCII /OR:/
1317 006637    040    045    117    .ASCII / %0/
1318 006642    066    045    116    .ASCII /6%N/
1319 006645    045    116    000    .ASCII /%N/<00>
1320 006650    045    101    107    P.AAU: .ASCII /%AG/
1321 006653    1,7    117    104    .ASCII /OOD/
1322 006656    040    116    111    .ASCII / NI/
1323 006661    102    040    104    .ASCII /B D/
1324 006664    101    124    101    .ASCII /ATA/
1325 006667    072    040    045    .ASCII /: %/
1326 006672    117    062    045    .ASCII /02%/
1327 006675    101    040    040    .ASCII /A /
1328 006700    040    040    102    .ASCII / B/
1329 006703    101    104    040    .ASCII /AD /
1330 006706    116    111    102    .ASCII /NIB/
1331 006711    140    104    101    .ASCII / DA/
1332 006714    124    101    072    .ASCII /TA:/
1333 006717    040    045    117    .ASCII / %0/
1334 006722    062    045    101    .ASCII /2%A/
1335 006725    040    040    040    .ASCII / /
1336 006730    040    116    111    .ASCII / NI/
1337 006733    102    040    120    .ASCII /B P/
1338 006736    117    123    072    .ASCII /OS:/
1339 006741    040    045    117    .ASCII / %0/
1340 006744    064    045    116    .ASCII /4%N/
1341 006747    045    116    000    .ASCII /%N/<00>
1342 006752    045    101    116    P.AAV: .ASCII /%AN/
1343 006755    .    102    040    .ASCII /IB /
1344 006760    1,1    116    040    .ASCII /IN /
1345 006763    105    122    122    .ASCII /ERR/
1346 006766    117    122    072    .ASCII /OR:/
1347 006771    040    045    104    .ASCII / %D/

```

```

1349      ;ML4AD
1350      ;
1351
1352 006774    064    045    116    .ASCII /4ZN/
1353 006777    045    116    000    .ASCII /ZN/<00>
1354 007002    045    101    106    P.AAW: .ASCII /ZAF/
1355 007005    101    111    114    .ASCII /AIL/
1356 007010    105    104    040    .ASCII /ED /
1357 007013    101    124    072    .ASCII /AT:/
1358 007016    040    045    117    .ASCII / 80/
1359 007021    066    045    116    .ASCII /6ZN/
1360 007024    045    116    000    .ASCII /ZN/<00>
1361 007027    000
1362 007030    045    101    122    P.AAX: .ASCII /ZAR/
1363 007033    105    120    114    .ASCII /EPL/
1364 007036    101    103    105    .ASCII /ACE/
1365 007041    040    101    122    .ASCII / AR/
1366 007044    122    040    115    .ASCII /R M/
1367 007047    117    104    072    .ASCII /OD:/
1368 007052    040    045    104    .ASCII / 8D/
1369 007055    062    045    116    .ASCII /2ZN/
1370 007060    045    116    000    .ASCII /ZN/<00>
1371 007063    000
1372 007064    045    101    106    P.AAY: .ASCII /ZAF/
1373 007067    101    111    114    .ASCII /AIL/
1374 007072    105    104    040    .ASCII /ED /
1375 007075    101    124    040    .ASCII /AT /
1376 007100    104    123    101    .ASCII /DSA/
1377 007103    072    040    045    .ASCII /: %/
1378 007106    117    066    045    .ASCII /06%/
1379 007111    116    045    116    .ASCII /NZN/
1380 007114    000    000
1381 007116    045    101    102    P.AAZ: .ASCII <00><00>
1382 007121    111    124    074    .ASCII /IT</
1383 007124    061    065    072    .ASCII /15:/
1384 007127    061    060    076    .ASCII /10>/
1385 007132    072    040    045    .ASCII /: %/
1386 007135    102    066    045    .ASCII /B6%/
1387 007140    101    040    040    .ASCII /A /
1388 007143    040    040    102    .ASCII / B/
1389 007146    111    124    074    .ASCII /IT</
1390 007151    071    072    060    .ASCII /9:0/
1391 007154    076    072    040    .ASCII />: /
1392 007157    045    102    061    .ASCII /XB1/
1393 007162    060    045    116    .ASCII /0ZN/
1394 007165    045    116    000    .ASCII /ZN/<00>
1395 007170    045    101    106    P.ABA: .ASCII /ZAF/
1396 007173    101    111    114    .ASCII /AIL/
1397 007176    111    116    107    .ASCII /ING/
1398 007201    040    122    105    .ASCII / RE/
1399 007204    107    040    101    .ASCII /G A/
1400 007207    104    122    123    .ASCII /DRS/
1401 007212    072    040    045    .ASCII /: %/
1402 007215    117    066    045    .ASCII /06%/
1403 007220    116    045    116    .ASCII /NZN/

```

```

1405      :ML4AD
1406      :
1407      :
1408 007223 000
1409 007224 045 101 106 P.ABB: .ASCII <00>
1410 007227 101 111 114 .ASCII /ZAF/
1411 007232 111 116 107 .ASCII /AIL/
1412 007235 040 106 125 .ASCII /ING/
1413 007240 116 103 072 .ASCII /FU/
1414 007243 040 045 117 .ASCII /NC:/
1415 007246 066 045 116 .ASCII /XO/
1416 007251 045 116 000 .ASCII /6ZN/
1417 007254 040 045 101 P.ABC: .ASCII /ZN/<00>
1418 007257 117 106 106 .ASCII /XA/
1419 007262 137 123 105 .ASCII /OFF/
1420 007265 124 040 103 .ASCII /SE/
1421 007270 116 124 040 .ASCII /T C/
1422 007273 106 117 122 .ASCII /NT /
1423 007276 040 116 111 .ASCII /FOR/
1424 007301 102 040 072 .ASCII /NI/
1425 007304 040 045 104 .ASCII /B :/
1426 007307 062 040 045 .ASCII /XD/
1427 007312 101 040 075 .ASCII /2 %/
1428 007315 040 045 104 .ASCII /A =/
1429 007320 062 040 045 .ASCII /XD/
1430 007323 116 045 116 .ASCII /2 %/
1431 007326 000 000 .ASCII /NZN/
1432 007330 045 101 127 P.ABD: .ASCII <00><00>
1433 007333 122 117 124 .ASCII /XAW/
1434 007336 105 072 040 .ASCII /ROT/
1435 007341 045 104 062 .ASCII /E: /
1436 007344 045 101 040 .ASCII /XD2/
1437 007347 040 040 040 .ASCII /XA /
1438 007352 122 105 101 .ASCII / /
1439 007355 104 072 040 .ASCII /REA/
1440 007360 045 104 062 .ASCII /D: /
1441 007363 045 116 045 .ASCII /XD2/
1442 007366 116 000 .ASCII /ZNZ/
1443 007370 045 101 116 P.ABE: .ASCII /N/<00>
1444 007373 111 102 102 .ASCII /XAN/
1445 007376 114 105 123 .ASCII /IBB/
1446 007401 040 130 106 .ASCII /LES/
1447 007404 105 122 105 .ASCII /XF/
1448 007407 104 040 102 .ASCII /ERE/
1449 007412 105 106 117 .ASCII /D B/
1450 007415 122 105 040 .ASCII /EFO/
1451 007420 105 122 122 .ASCII /RE /
1452 007423 117 122 072 .ASCII /ERR/
1453 007426 040 045 104 .ASCII /OR:/
1454 007431 063 045 116 .ASCII /XD/
1455 007434 000 000 .ASCII /3ZN/
1456 007436 045 101 106 P.ABF: .ASCII <00><00>
1457 007441 101 111 114 .ASCII /ZAF/
1458 007444 111 116 107 .ASCII /AIL/
1459 007447 040 122 105 .ASCII /ING/
        .ASCII / RE/

```

```

1461      :ML4AD
1462      :
1463
1464 007452    107    072    040    .ASCII /G: /
1465 007455    045    117    066    .ASCII /%06/
1466 007460    045    101    040    .ASCII /%A /
1467 007463    107    117    117    .ASCII /G00/
1468 007466    104    040    104    .ASCII /D D/
1469 007471    101    124    101    .ASCII /ATA/
1470 007474    072    040    045    .ASCII /: %/
1471 007477    117    066    045    .ASCII /06%/
1472 007502    101    040    102    .ASCII /A B/
1473 007505    101    104    040    .ASCII /AD /
1474 007510    104    101    124    .ASCII /DAT/
1475 007513    101    072    040    .ASCII /A: /
1476 007516    045    117    066    .ASCII /%06/
1477 007521    045    116    045    .ASCII /%N%/
1478 007524    116    000    .ASCII /N/<00>
1479 007526    045    116    045 P.ABG: .ASCII /%N%/
1480 007531    101    104    111    .ASCII /ADI/
1481 007534    101    107    116    .ASCII /AGN/
1482 007537    117    123    111    .ASCII /OSI/
1483 007542    116    107    040    .ASCII /NG /
1484 007545    125    116    111    .ASCII /UNI/
1485 007550    124    040    045    .ASCII /T %/
1486 007553    117    061    045    .ASCII /01%/
1487 007556    116    045    116    .ASCII /%N%/
1488 007561    000    .ASCII <00>
1489 007562    045    101    124 P.ABH: .ASCII /%AT/
1490 007565    111    115    105    .ASCII /IME/
1491 007570    104    040    117    .ASCII /D O/
1492 007573    125    124    040    .ASCII /UT /
1493 007576    104    125    122    .ASCII /DUR/
1494 007601    111    116    107    .ASCII /ING/
1495 007604    040    115    102    .ASCII / MB/
1496 007607    125    123    040    .ASCII /US /
1497 007612    045    117    062    .ASCII /%02/
1498 007615    045    101    040    .ASCII /%A /
1499 007620    106    125    116    .ASCII /FUN/
1500 007623    103    045    116    .ASCII /%CN/
1501 007626    045    116    000    .ASCII /%N/<00>
1502 007631    000    .ASCII <00>
1503 007632    045    101    103 P.ABI: .ASCII /%AC/
1504 007635    122    103    040    .ASCII /RC /
1505 007640    107    105    116    .ASCII /GEN/
1506 007643    040    075    040    .ASCII / = /
1507 007646    102    072    040    .ASCII /B: /
1508 007651    045    102    066    .ASCII /%B6/
1509 007654    045    101    040    .ASCII /%A /
1510 007657    101    072    040    .ASCII /A: /
1511 007662    045    102    066    .ASCII /%B6/
1512 007665    045    101    040    .ASCII /%A /
1513 007670    120    072    040    .ASCII /P: /
1514 007673    045    102    066    .ASCII /%B6/
1515 007676    045    116    000    .ASCII /%N/<00>

```

1517									
1518									
1519									
1520	007701	000							
1521	007702	045	101	103	P.ABJ:	.ASCII	<00>		
1522	007703	122	103	040		.ASCII	/XAC/		
1523	007710	103	101	114		.ASCII	/RC /		
1524	007713	040	075	040		.ASCII	/CAL/		
1525	007716	102	072	040		.ASCII	/ = /		
1526	007721	045	102	066		.ASCII	/B: /		
1527	007724	045	101	040		.ASCII	/XB6/		
1528	007727	101	072	040		.ASCII	/ZA /		
1529	007732	045	102	066		.ASCII	/A: /		
1530	007735	045	101	040		.ASCII	/XB6/		
1531	007740	120	072	040		.ASCII	/ZA /		
1532	007743	045	102	066		.ASCII	/P: /		
1533	007746	045	116	045		.ASCII	/XB6/		
1534	007751	116	000	000		.ASCII	/ZNZ/		
1535	007754	045	101	106	P.ABK:	.ASCII	/N/<00><00>		
1536	007757	101	111	114		.ASCII	/XAF/		
1537	007762	105	104	040		.ASCII	/AIL/		
1538	007765	101	124	040		.ASCII	/ED /		
1539	007770	120	114	117		.ASCII	/AT /		
1540	007773	107	072	040		.ASCII	/PLO/		
1541	007776	045	104	062		.ASCII	/G: /		
1542	010001	045	101	040		.ASCII	/XD2/		
1543	010004	103	110	101		.ASCII	/XA /		
1544	010007	116	116	105		.ASCII	/CHA/		
1545	010012	114	072	040		.ASCII	/NNE/		
1546	010015	045	104	062		.ASCII	/L: /		
1547	010020	045	116	045		.ASCII	/XD2/		
1548	010023	116	000	000		.ASCII	/ZNZ/		
1549	010026	045	101	106	P.ABL:	.ASCII	/N/<00><00>		
1550	010031	101	111	114		.ASCII	/XAF/		
1551	010034	105	104	040		.ASCII	/AIL/		
1552	010037	101	124	040		.ASCII	/ED /		
1553	010042	127	122	104		.ASCII	/AT /		
1554	010045	072	040	045		.ASCII	/WRD/		
1555	010050	104	062	045		.ASCII	/: X/		
1556	010053	101	040	102		.ASCII	/D2X/		
1557	010056	111	124	072		.ASCII	/A B/		
1558	010061	040	045	104		.ASCII	/IT:/		
1559	010064	062	045	116		.ASCII	/ XD/		
1560	010067	045	116	000		.ASCII	/2XN/		
1561	010072	045	101	122	P.ABM:	.ASCII	/XN/<00>		
1562	010075	105	107	111		.ASCII	/XAR/		
1563	010100	123	124	105		.ASCII	/EGI/		
1564	010103	122	045	123		.ASCII	/STE/		
1565	010106	063	045	101		.ASCII	/RZS/		
1566	010111	101	104	104		.ASCII	/3XA/		
1567	010114	122	105	123		.ASCII	/ADD/		
1568	010117	123	045	123		.ASCII	/RES/		
1569	010122	067	045	101		.ASCII	/SZS/		
1570	010125	103	117	116		.ASCII	/7XA/		
1571	010130	124	105	116		.ASCII	/CON/		
						.ASCII	/TEN/		

Line	Code	Value 1	Value 2	Value 3	Label	Declaration
1573					:ML4AD	
1574					:	
1575						ROU/INE DECLARATION SECTION
1576	010133	124	123	045		.ASCII /TS%/
1577	010136	116	045	116		.ASCII /NZN/
1578	010141	000				.ASCII <00>
1579	010142	045	123	045	P.ABN:	.ASCII /XS%/
1580	010145	124	045	123		.ASCII /TXS/
1581	010150	065	045	117		.ASCII /5X0/
1582	010153	066	045	123		.ASCII /6XS/
1583	010156	064	045	117		.ASCII /4X0/
1584	010161	061	066	045		.ASCII /16X/
1585	010164	116	000			.ASCII /N/<00>
1586	010166	045	123	064	P.ABO:	.ASCII /XS4/
1587	010171	045	124	045		.ASCII /XTZ/
1588	010174	101	072	045		.ASCII /A:Z/
1589	010177	123	045	117		.ASCII /SX0/
1590	010202	066	045	116		.ASCII /6XN/
1591	010205	000				.ASCII <00>
1592	010206	045	123	064	P.ABP:	.ASCII /XS4/
1593	010211	045	124	045		.ASCII /XTZ/
1594	010214	101	072	045		.ASCII /A:Z/
1595	010217	123	045	104		.ASCII /SX0/
1596	010222	061	045	104		.ASCII /1XD/
1597	010225	061	045	104		.ASCII /1XD/
1598	010230	061	045	104		.ASCII /1XD/
1599	010233	061	045	116		.ASCII /1XN/
1600	010236	000	000			.ASCII <00><00>
1601	010240	045	124	045	P.ABO:	.ASCII /XTZ/
1602	010243	116	000	000		.ASCII /N/<00><00>
1603	010246	045	124	045	P.ABR:	.ASCII /XTZ/
1604	010251	124	045	116		.ASCII /TXN/
1605	010254	000	000			.ASCII <00><00>
1606	010256	045	124	045	P.ABS:	.ASCII /XTZ/
1607	010261	124	045	124		.ASCII /TXT/
1608	010264	045	116	000		.ASCII /XN/<00>
1609	010267	000				.ASCII <00>
1610	010270	045	124	045	P.ABT:	.ASCII /XTZ/
1611	010273	124	045	124		.ASCII /TXT/
1612	010276	045	124	045		.ASCII /XTZ/
1613	010301	116	000	000		.ASCII /N/<00><00>
1614	010304	045	124	045	P.ABU:	.ASCII /XTZ/
1615	010307	124	045	124		.ASCII /TXT/
1616	010312	045	124	045		.ASCII /XTZ/
1617	010315	124	045	116		.ASCII /TXN/
1618	010320	000	000			.ASCII <00><00>
1619	010322	045	124	045	P.ABV:	.ASCII /XTZ/
1620	010325	124	045	124		.ASCII /TXT/
1621	010330	045	124	045		.ASCII /XTZ/
1622	010333	124	045	124		.ASCII /TXT/
1623	010336	045	116	000		.ASCII /XN/<00>
1624	010341	000				.ASCII <00>
1625	010342	045	124	045	P.ABW:	.ASCII /XTZ/
1626	010345	124	045	124		.ASCII /TXT/
1627	010350	045	124	045		.ASCII /XTZ/


```
1629      :ML4AD
1630      :
1631
1632 010353 124 045 124 .ASCII /T%T/
1633 010356 045 124 045 .ASCII /%T%/
1634 010361 116 000 000 .ASCII /N/<00><00>
1635 010364 045 124 045 P.ABX: .ASCII /%T%/
1636 010367 124 045 124 .ASCII /T%T/
1637 010372 045 124 045 .ASCII /%T%/
1638 010375 124 045 124 .ASCII /T%T/
1639 010400 045 124 045 .ASCII /%T%/
1640 010403 124 045 116 .ASCII /T%N/
1641 010406 000 000 .ASCII <00><00>
1642 010410 045 124 045 P.ABY: .ASCII /%T%/
1643 010413 124 045 124 .ASCII /T%T/
1644 010416 045 124 045 .ASCII /%T%/
1645 010421 124 045 124 .ASCII /T%T/
1646 010424 045 124 045 .ASCII /%T%/
1647 010427 124 045 124 .ASCII /T%T/
1648 010432 045 116 000 .ASCII /%N/<00>
1649 010435 000 .ASCII <00>
1650 010436 045 124 045 P.ABZ: .ASCII /%T%/
1651 010441 124 045 124 .ASCII /T%T/
1652 010444 045 124 045 .ASCII /%T%/
1653 010447 124 045 124 .ASCII /T%T/
1654 010452 045 124 045 .ASCII /%T%/
1655 010455 124 045 124 .ASCII /T%T/
1656 010460 045 124 045 .ASCII /%T%/
1657 010463 116 000 000 .ASCII /N/<00><00>
1658 010466 045 124 045 P.ACA: .ASCII /%T%/
1659 010471 124 045 124 .ASCII /T%T/
1660 010474 045 124 045 .ASCII /%T%/
1661 010477 124 045 124 .ASCII /T%T/
1662 010502 045 124 045 .ASCII /%T%/
1663 010505 124 045 124 .ASCII /T%T/
1664 010510 045 124 045 .ASCII /%T%/
1665 010513 124 045 116 .ASCII /T%N/
1666 010516 000 000 .ASCII <00><00>
1667 010520 040 107 117 P.ACB: .ASCII / GO/
1668 010523 000 .ASCII <00>
1669 010524 040 104 122 P.ACC: .ASCII / DR/
1670 010527 126 137 122 .ASCII /V R/
1671 010532 104 131 000 .ASCII /D%/<00>
1672 010535 000 .ASCII <00>
1673 010536 040 111 114 P.ACD: .ASCII / IL/
1674 010541 106 000 000 .ASCII /F/<00><00>
1675 010544 040 117 120 P.ACE: .ASCII / OP/
1676 010547 111 000 000 .ASCII /I/<00><00>
1677 010552 040 102 101 P.ACF: .ASCII / BA/
1678 010555 104 000 000 .ASCII /D/<00><00>
1679 010560 040 107 117 P.ACG: .ASCII / GO/
1680 010563 117 104 000 .ASCII /OD/<00>
1681 010566 040 120 101 P.ACH: .ASCII / PA/
1682 010571 122 111 124 .ASCII /RIT/
1683 010574 131 040 116 .ASCII /Y N/
```

```

1685      :ML4AD
1686      :
1687      :
1688 010577 117 124 000
1689 010602 040 107 105 P.ACI: .ASCII /OT/<00>
1690 010605 116 105 122 .ASCII /GE/
1691 010610 101 124 105 .ASCII /NER/
1692 010613 104 000 000 .ASCII /ATE/
1693 010616 040 104 105 P.ACJ: .ASCII /D/<00><00>
1694 010621 124 105 103 .ASCII /DE/
1695 010624 124 105 104 .ASCII /TEC/
1696 010627 000 .ASCII /TED/
1697 010630 040 105 122 P.ACK: .ASCII <00>
1698 010633 122 117 122 .ASCII /ER/
1699 010636 000 000 .ASCII /ROR/
1700 010640 040 101 106 P.ACL: .ASCII <00><00>
1701 010643 124 105 122 .ASCII /AF/
1702 010646 000 000 .ASCII /TER/
1703 010650 040 104 125 P.ACM: .ASCII <00><00>
1704 010653 122 111 116 .ASCII /DU/
1705 010656 107 000 .ASCII /RIN/
1706 010660 040 101 124 P.ACN: .ASCII /G/<00>
1707 010663 000 .ASCII /AT/
1708 010664 040 106 101 P.ACO: .ASCII <00>
1709 010667 111 114 125 .ASCII /FA/
1710 010672 122 105 000 .ASCII /ILU/
1711 010675 000 .ASCII /RE/<00>
1712 010676 040 101 124 P.ACP: .ASCII <00>
1713 010701 101 000 000 .ASCII /AT/
1714 010704 040 101 124 P.ACQ: .ASCII /A/<00><00>
1715 010707 124 116 000 .ASCII /AT/
1716 010712 040 127 122 P.ACR: .ASCII /TN/<00>
1717 010715 111 124 111 .ASCII /WR/
1718 010720 116 107 000 .ASCII /ITI/
1719 010723 000 .ASCII /NG/<00>
1720 010724 040 126 126 P.ACS: .ASCII <00>
1721 010727 000 .ASCII /VV/
1722 010730 040 106 125 P.ACT: .ASCII <00>
1723 010733 116 103 000 .ASCII /FU/
1724 010736 040 124 122 P.ACU: .ASCII /NC/<00>
1725 010741 105 000 000 .ASCII /TR/
1726 010744 040 122 115 P.ACV: .ASCII /E/<00><00>
1727 010747 122 000 000 .ASCII /RM/
1728 010752 040 105 130 P.ACW: .ASCII /R/<00><00>
1729 010755 103 105 123 .ASCII /EX/
1730 010760 123 111 126 .ASCII /CES/
1731 010763 105 000 000 .ASCII /SIV/
1732 010766 040 115 102 P.ACX: .ASCII /E/<00><00>
1733 010771 125 123 000 .ASCII /MB/
1734 010774 040 104 101 P.ACY: .ASCII /US/<00>
1735 010777 124 101 000 .ASCII /DA/
1736 011002 040 103 117 P.ACZ: .ASCII /TA/<00>
1737 011005 116 124 111 .ASCII /CO/
1738 011010 116 125 111 .ASCII /NTI/
1739 011013 124 131 000 .ASCII /NUI/
        .ASCII /TY/<00>

```

```

1741      :ML4AD
1742      :
1743
1744 011016 040 101 117 P.ADA: .ASCII / AD/
1745 011021 105 000 000 .ASCII /E/<00><00>
1746 011024 040 114 102 P.ADB: .ASCII / LB/
1747 011027 124 000 000 .ASCII /T/<00><00>
1748 011032 040 120 122 P.ADC: .ASCII / PR/
1749 011035 105 115 101 .ASCII /EMA/
1750 011040 124 125 122 .ASCII /TUR/
1751 011043 114 131 000 .ASCII /LY/<00>
1752 011046 040 111 101 P.ADD: .ASCII / IA/
1753 011051 105 000 000 .ASCII /E/<00><00>
1754 011054 040 111 116 P.ADE: .ASCII / IN/
1755 011057 103 122 105 .ASCII /CRE/
1756 011062 115 105 116 .ASCII /MEN/
1757 011065 124 000 000 .ASCII /T/<00><00>
1758 011070 040 127 111 P.ADF: .ASCII / WI/
1759 011073 124 110 000 .ASCII /TH/<00>
1760 011076 040 125 126 P.ADG: .ASCII / UV/
1761 011101 000 .ASCII <00>
1762 011102 040 125 116 P.ADH: .ASCII / UN/
1763 011105 123 000 000 .ASCII /S/<00><00>
1764 011110 040 120 122 P.ADI: .ASCII / PR/
1765 011113 117 115 000 .ASCII /OM/<00>
1766 011116 040 117 122 P.ADJ: .ASCII / OR/
1767 011121 000 .ASCII <00>
1768 011122 040 123 105 P.ADK: .ASCII / SE/
1769 011125 114 105 103 .ASCII /LEC/
1770 011130 124 000 .ASCII /T/<00>
1771 011132 040 122 105 P.ADL: .ASCII / RE/
1772 011135 107 000 000 .ASCII /G/<00><00>
1773 011140 040 125 116 P.ADM: .ASCII / UN/
1774 011143 111 121 125 .ASCII /IQU/
1775 011146 105 000 .ASCII /E/<00>
1776 011150 040 061 064 P.ADN: .ASCII / 14/
1777 011153 000 .ASCII <00>
1778 011154 040 116 111 P.ADO: .ASCII / NI/
1779 011157 102 102 114 .ASCII /BBL/
1780 011162 105 040 103 .ASCII /E C/
1781 011165 116 124 000 .ASCII /NT/<00>
1782 011170 040 107 124 P.ADP: .ASCII / GT/
1783 011173 122 000 000 .ASCII /R/<00><00>
1784 011176 040 127 110 P.ADQ: .ASCII / WH/
1785 011201 111 114 105 .ASCII /ILE/
1786 011204 000 000 .ASCII <00><00>
1787 011206 040 124 122 P.ADR: .ASCII / TR/
1788 011211 105 000 000 .ASCII /E/<00><00>
1789 011214 040 111 116 P.ADS: .ASCII / IN/
1790 011217 111 124 111 .ASCII /ITI/
1791 011222 101 114 000 .ASCII /AL/<00>
1792 011225 000 .ASCII <00>
1793 011226 040 117 106 P.ADT: .ASCII / OF/
1794 011231 106 137 123 .ASCII /F S/
1795 011234 105 124 000 .ASCII /ET/<00>

```

```

1797
1798
1799
1800 011237 000
1801 011240 040 103 117 P.ADU: .ASCII <00>
1802 011243 125 116 124 .ASCII / CO/
1803 011246 000 000 .ASCII /UNT/
1804 011250 040 104 105 P.ADV: .ASCII <00><00>
1805 011253 114 101 131 .ASCII / DE/
1806 011256 000 000 .ASCII /LAY/
1807 011260 040 124 105 P.ADW: .ASCII <00><00>
1808 011263 123 124 123 .ASCII / TE/
1809 011266 000 000 .ASCII /STS/
1810 011270 040 101 104 P.ADX: .ASCII <00><00>
1811 011273 122 123 000 .ASCII / AD/
1812 011276 040 103 117 P.ADY: .ASCII /RS/<00>
1813 011301 125 116 124 .ASCII / CO/
1814 011304 105 122 000 .ASCII /UNT/
1815 011307 000 .ASCII /ER/<00>
1816 011310 040 122 105 P.ADZ: .ASCII <00>
1817 011313 107 000 000 .ASCII / RE/
1818 011316 040 124 105 P.AEA: .ASCII /G/<00><00>
1819 011321 123 124 105 .ASCII / TE/
1820 011324 104 000 .ASCII /STE/
1821 011326 040 116 111 P.AEB: .ASCII /D/<00>
1822 011331 102 102 114 .ASCII / NI/
1823 011334 105 000 .ASCII /BBL/
1824 011336 040 101 114 P.AEC: .ASCII /E/<00>
1825 011341 114 000 000 .ASCII / AL/
1826 011344 040 124 105 P.AED: .ASCII /L/<00><00>
1827 011347 123 124 000 .ASCII / TE/
1828 011352 040 130 106 P.AEE: .ASCII /ST/<00>
1829 011355 105 122 105 .ASCII / XF/
1830 011360 104 000 .ASCII /ERE/
1831 011362 040 116 111 P.AEF: .ASCII /D/<00>
1832 011365 102 102 114 .ASCII / NI/
1833 011370 105 125 000 .ASCII /BBL/
1834 011373 000 .ASCII /ES/<00>
1835 011374 040 123 103 P.AEG: .ASCII <00>
1836 011377 000 .ASCII / SC/
1837 011400 040 115 125 P.AEH: .ASCII <00>
1838 011403 114 124 111 .ASCII / MU/
1839 011406 120 114 105 .ASCII /LTI/
1840 011411 130 105 122 .ASCII /PLE/
1841 011414 000 000 .ASCII /XER/
1842 011416 040 125 116 P.AEI: .ASCII <00><00>
1843 011421 105 130 120 .ASCII / UN/
1844 011424 105 103 124 .ASCII /EXP/
1845 011427 105 104 000 .ASCII /ECT/
1846 011432 040 116 105 P.AEJ: .ASCII /ED/<00>
1847 011435 104 000 000 .ASCII / NE/
1848 011440 040 111 114 P.AEK: .ASCII /D/<00><00>
1849 011443 122 000 000 .ASCII / IL/
1850 011446 040 103 122 P.AEL: .ASCII /R/<00><00>
1851 011451 103 000 000 .ASCII / CR/
        .ASCII /C/<00><00>

```

```

1853      :ML4AD
1854      :
1855      :
1856 011454 040 123 107 P.AEM: .ASCII / SG/
1857 011457 114 000 000 .ASCII /L/<00><00>
1858 011462 040 105 103 P.AEN: .ASCII / EC/
1859 011465 110 000 000 .ASCII /H/<00><00>
1860 011470 125 116 103 P.AEO: .ASCII /UNC/
1861 011473 000 .ASCII <00>
1862 011474 040 102 111 P.AEP: .ASCII / BI/
1863 011477 124 000 000 .ASCII /T/<00><00>
1864 011502 040 103 110 P.AEQ: .ASCII / CH/
1865 011505 101 116 116 .ASCII /ANN/
1866 011510 105 114 000 .ASCII /EL/<00>
1867 011513 000 .ASCII <00>
1868 011514 040 114 101 P.AER: .ASCII / LA/
1869 011517 124 103 110 .ASCII /TCH/
1870 011522 000 000 .ASCII <00><00>
1871 011524 040 104 103 P.AES: .ASCII / DC/
1872 011527 113 000 000 .ASCII /K/<00><00>
1873 011532 040 102 125 P.AET: .ASCII / BU/
1874 011535 123 000 000 .ASCII /S/<00><00>
1875 011540 040 123 131 P.AEU: .ASCII / SY/
1876 011543 116 104 122 .ASCII /NDR/
1877 011546 117 116 105 .ASCII /ONE/
1878 011551 000 .ASCII <00>
1879 011552 040 104 105 P.AEV: .ASCII / DE/
1880 011555 103 117 104 .ASCII /COD/
1881 011560 105 000 .ASCII /E/<00>
1882 011562 040 116 117 P.AEW: .ASCII / NO/
1883 011565 124 000 000 .ASCII /T/<00><00>
1884 011570 040 125 116 P.AEX: .ASCII / UN/
1885 011573 103 117 122 .ASCII /COR/
1886 011576 122 105 103 .ASCII /REC/
1887 011601 124 101 102 .ASCII /TAB/
1888 011604 114 105 000 .ASCII /LE/<00>
1889 011607 000 .ASCII <00>
1890 011610 040 102 111 P.AEY: .ASCII / BI/
1891 011613 124 040 116 .ASCII /T N/
1892 011616 117 124 040 .ASCII /OT /
1893 011621 123 105 124 .ASCII /SET/
1894 011624 000 000 .ASCII <00><00>
1895 011626 040 102 111 P.AEZ: .ASCII / BI/
1896 011631 124 040 116 .ASCII /T N/
1897 011634 117 124 040 .ASCII /OT /
1898 011637 103 114 122 .ASCII /CLR/
1899 011642 000 000 .ASCII <00><00>
1900 011644 040 116 117 P.AFA: .ASCII / NO/
1901 011647 040 122 105 .ASCII / RE/
1902 011652 123 120 117 .ASCII /SPO/
1903 011655 116 103 105 .ASCII /NCE/
1904 011660 040 101 106 .ASCII / AF/
1905 011663 124 105 122 .ASCII /TER/
1906 011666 040 061 056 .ASCII / 1./
1907 011671 065 040 125 .ASCII /5 U/

```

```

1909      :ML4AD
1910      :
1911      :
1912 011674 123 000
1913 011676 040 104 101 P.AFB: .ASCII /S/<00>
1914 011701 124 101 040 .ASCII / DA/
1915 011704 105 122 122 .ASCII /TA /
1916 011707 117 122 123 .ASCII /ERR/
1917 011712 000 000 .ASCII /ORS/
1918 011714 040 102 111 P.AFC: .ASCII <00><00>
1919 011717 124 040 123 .ASCII / BI/
1920 01172 105 124 000 .ASCII /T S/
1921 01172 000 .ASCII /ET/<00>
1922 01172 000 .ASCII <00>
1923 011731 124 040 102 111 P.AFD: .ASCII / BI/
1924 011734 114 122 000 .ASCII /T C/
1925 011737 000 .ASCII /LR/<00>
1926 011740 040 117 106 P.AFE: .ASCII <00>
1927 011743 040 117 124 .ASCII / OF/
1928 011746 110 105 122 .ASCII / OT/
1929 011751 040 104 122 .ASCII /HER/
1930 011754 111 126 105 .ASCII / DR/
1931 011757 123 000 000 .ASCII /IVE/
1932 011762 040 103 114 P.AFF: .ASCII /S/<00><00>
1933 011765 101 123 123 .ASCII / CL/
1934 011770 040 101 000 .ASCII /ASS/
1935 011773 000 .ASCII / A/<00>
1936 011774 040 103 114 P.AFG: .ASCII <00>
1937 011777 101 123 123 .ASCII / CL/
1938 012002 040 102 000 .ASCII /ASS/
1939 012005 000 .ASCII / B/<00>
1940 012006 040 124 117 P.AFH: .ASCII <00>
1941 012011 040 106 111 .ASCII / TO/
1942 012014 116 104 000 .ASCII / FI/
1943 012017 000 .ASCII /ND/<00>
1944 012020 040 116 117 P.AFI: .ASCII <00>
1945 012023 124 040 114 .ASCII / NO/
1946 012026 101 124 103 .ASCII /T L/
1947 012031 110 105 104 .ASCII /ATC/
1948 012034 000 000 .ASCII /HED/
1949 012036 040 123 111 P.AFJ: .ASCII <00><00>
1950 012041 116 107 114 .ASCII / SI/
1951 012044 105 040 102 .ASCII /NGL/
1952 012047 111 124 040 .ASCII /E B/
1953 012052 105 122 122 .ASCII /IT /
1954 012055 117 122 000 .ASCII /ERR/
1955 012060 040 115 125 P.AFK: .ASCII /OR/<00>
1956 012063 114 124 111 .ASCII / MU/
1957 012066 120 114 105 .ASCII /LTI/
1958 012071 040 102 111 .ASCII /PLE/
1959 012074 124 040 105 .ASCII / BI/
1960 012077 122 122 117 .ASCII /T E/
1961 012102 122 000 .ASCII /RRO/
1962 012104 040 040 040 P.AFL: .ASCII /R/<00>
1963 012107 040 040 040 .ASCII / /

```

				:ML4AD			
				:	ROUTINE DECLARATION SECTION		
1965							
1966							
1967							
1968	012112	040	040		.ASCII	/ /	
1969	012115	122	105		.ASCII	/REG/	
1970	012120	111	123		.ASCII	/IST/	
1971	012123	105	122		.ASCII	/ER /	
1972	012126	040	040		.ASCII	/ D/	
1973	012131	125	115		.ASCII	/UMP/	
1974	012134	000	000		.ASCII	<00><00>	
1975	012136	123	105	122	P.AFM:	.ASCII	/SER/
1976	012141	111	101		.ASCII	/IAL/	
1977	012144	040	043	000	.ASCII	/ #/<00>	
1978	012147	000			.ASCII	<00>	
1979	012150	040	115	105	P.AFN:	.ASCII	/ ME/
1980	012153	115	040		.ASCII	/M S/	
1981	012156	111	132		.ASCII	/IZI/	
1982	012161	116	107	000	.ASCII	/NG/<00>	
1983	012164	040	116	117	P.AFO:	.ASCII	/ NO/
1984	012167	117	120	000	.ASCII	/OP/<00>	
1985	012172	040	104	122	P.AFP:	.ASCII	/ DR/
1986	012175	126	000	000	.ASCII	/V/<00><00>	
1987	012200	040	127	122	P.AFQ:	.ASCII	/ WR/
1988	012203	111	124	105	.ASCII	/ITE/	
1989	012206	040	103	110	.ASCII	/ CH/	
1990	012211	105	103	113	.ASCII	/ECK/	
1991	012214	000	000		.ASCII	<00><00>	
1992	012216	040	127	122	P.AFR:	.ASCII	/ WR/
1993	012221	111	124	105	.ASCII	/ITE/	
1994	012224	000	000		.ASCII	<00><00>	
1995	012226	040	122	105	P.AFS:	.ASCII	/ RE/
1996	012231	101	104	000	.ASCII	/AD/<00>	
1997	012234	040	103	114	P.AFT:	.ASCII	/ CL/
1998	012237	105	101	122	.ASCII	/EAR/	
1999	012242	000	000		.ASCII	<00><00>	
2000	012244	040	103	117	P.AFU:	.ASCII	/ CO/
2001	012247	115	120	040	.ASCII	/MP /	
2002	012252	105	122	122	.ASCII	/ERR/	
2003	012255	117	122	000	.ASCII	/OR/<00>	
2004	012260	040	123	131	P.AFV:	.ASCII	/ SY/
2005	012263	123	040	103	.ASCII	/S C/	
2006	012266	114	122	000	.ASCII	/LR/<00>	
2007	012271	000			.ASCII	<00>	
2008	012272	040	123	105	P.AFW:	.ASCII	/ SE/
2009	012275	101	122	103	.ASCII	/ARC/	
2010	012300	110	000		.ASCII	/H/<00>	
2011	012302	040	122	105	P.AFX:	.ASCII	/ RE/
2012	012305	101	104	055	.ASCII	/AD-/	
2013	012310	111	116	055	.ASCII	/IN-/	
2014	012313	120	122	105	.ASCII	/PRE/	
2015	012316	123	105	124	.ASCII	/SET/	
2016	012321	000			.ASCII	<00>	
2017	012322	040	111	114	P.AFY:	.ASCII	/ IL/
2018	012325	114	105	107	.ASCII	/LEG/	
2019	012330	101	114	000	.ASCII	/AL/<00>	


```
2021      :ML4AD
2022      :
2023      :
2024 012333 000
2025 012334 040 101 102 P.AFZ: .ASCII <00>
2026 012337 117 122 124 .ASCII / AB/
2027 012342 000 000 .ASCII /ORT/
2028 012344 040 101 122 P.AGA: .ASCII <00><00>
2029 012347 122 040 122 .ASCII / AR/
2030 012352 104 137 127 .ASCII /R R/
2031 012355 122 124 000 .ASCII /D W/
2032 012360 040 107 117 P.AGB: .ASCII /RT/<00>
2033 012363 117 104 040 .ASCII / GO/
2034 012366 102 114 113 .ASCII /OD /
2035 012371 000 .ASCII /BLK/
2036 012372 040 122 105 P.AGC: .ASCII <00>
2037 012375 106 122 105 .ASCII / RE/
2038 012400 123 110 000 .ASCII /FRE/
2039 012403 000 .ASCII /SH/<00>
2040 012404 040 101 122 P.AGD: .ASCII <00>
2041 012407 122 101 131 .ASCII / AR/
2042 012412 000 000 .ASCII /RAY/
2043 012414 040 122 101 P.AGE: .ASCII <00><00>
2044 012417 115 055 102 .ASCII / RA/
2045 012422 125 123 000 .ASCII /M-B/
2046 012425 000 .ASCII /US/<00>
2047 012426 040 117 126 P.AGF: .ASCII <00>
2048 012431 105 122 106 .ASCII / OV/
2049 012434 114 117 127 .ASCII /ERF/
2050 012437 000 .ASCII /LOW/
2051 012440 040 103 110 P.AGG: .ASCII <00>
2052 012443 113 137 123 .ASCII / CH/
2053 012446 125 115 000 .ASCII /K S/
2054 012451 000 .ASCII /UM/<00>
2055 012452 040 114 101 P.AGH: .ASCII <00>
2056 012455 123 124 040 .ASCII / LA/
2057 012460 102 114 113 .ASCII /ST /
2058 012463 000 .ASCII /BLK/
2059 012464 040 111 116 P.AGI: .ASCII <00>
2060 012467 111 124 111 .ASCII / IN/
2061 012472 101 114 111 .ASCII /ITI/
2062 012475 132 105 000 .ASCII /ALI/
2063 012500 040 115 114 P.AGJ: .ASCII /ZE/<00>
2064 012503 103 123 061 .ASCII / ML/
2065 012506 000 000 .ASCII /CS1/
2066 012510 040 115 114 P.AGK: .ASCII <00><00>
2067 012513 104 123 000 .ASCII / ML/
2068 012516 040 115 114 P.AGL: .ASCII /DS/<00>
2069 012521 105 122 000 .ASCII / ML/
2070 012524 040 115 114 P.AGM: .ASCII /ER/<00>
2071 012527 115 122 000 .ASCII / ML/
2072 012532 040 115 114 P.AGN: .ASCII /MR/<00>
2073 012535 101 123 000 .ASCII / ML/
2074 012540 040 115 114 P.AGO: .ASCII /AS/<00>
2075 012543 104 101 000 .ASCII / ML/
2075 012543 104 101 000 .ASCII /DA/<00>
```

```

2077      :ML4AD
2078      :
2079      :
2080 012546 040 115 114 P.AGP: .ASCII / ML/
2081 012551 104 124 000 .ASCII /DT/<00>
2082 012554 040 115 114 P.AGQ: .ASCII / ML/
2083 012557 120 101 000 .ASCII /PA/<00>
2084 012562 040 115 114 P.AGR: .ASCII / ML/
2085 012565 123 116 000 .ASCII /SN/<00>
2086 012570 040 115 114 P.AGS: .ASCII / ML/
2087 012573 105 061 000 .ASCII /E1/<00>
2088 012576 040 115 114 P.AGT: .ASCII / ML/
2089 012601 105 062 000 .ASCII /E2/<00>
2090 012604 040 115 114 P.AGU: .ASCII / ML/
2091 012607 104 061 000 .ASCII /D1/<00>
2092 012612 040 115 114 P.AGV: .ASCII / ML/
2093 012615 104 062 000 .ASCII /D2/<00>
2094 012620 040 115 114 P.AGW: .ASCII / ML/
2095 012623 105 105 000 .ASCII /EE/<00>
2096 012626 040 115 114 P.AGX: .ASCII / ML/
2097 012631 105 114 000 .ASCII /EL/<00>
2098 012634 040 115 114 P.AGY: .ASCII / ML/
2099 012637 120 104 000 .ASCII /PD/<00>
2100 012642 040 115 114 P.AGZ: .ASCII / ML/
2101 012645 103 123 062 .ASCII /CS2/
2102 012650 000 000 .ASCII <00><00>
2103 012652 040 115 114 P.AHA: .ASCII / ML/
2104 012655 127 103 000 .ASCII /WC/<00>
2105 012660 040 115 114 P.AHB: .ASCII / ML/
2106 012663 102 101 000 .ASCII /BA/<00>
2107 012666 040 115 114 P.AHC: .ASCII / ML/
2108 012671 102 101 111 .ASCII /BAI/
2109 012674 000 000 .ASCII <00><00>
2110 012676 040 115 114 P.AHD: .ASCII / ML/
2111 012701 103 123 063 .ASCII /CS3/
2112 012704 000 000 .ASCII <00><00>
2113 012706 101 123 131 P.AHE: .ASCII /ASY/
2114 012711 116 103 110 .ASCII /NCH/
2115 012714 122 117 116 .ASCII /RON/
2116 012717 117 125 123 .ASCII /OUS/
2117 012722 040 106 101 .ASCII / FA/
2118 012725 111 114 125 .ASCII /ILU/
2119 012730 122 105 040 .ASCII /RE /
2120 012733 115 117 104 .ASCII /MOD/
2121 012736 125 114 105 .ASCII /ULE/
2122 012741 040 067 063 .ASCII / 73/
2123 012744 066 061 000 .ASCII /61/<00>
2124 012747 000 .ASCII <00>
2125 012750 123 131 116 P.AHF: .ASCII /SYN/
2126 012753 103 110 122 .ASCII /CHR/
2127 012756 117 116 117 .ASCII /ONO/
2128 012761 125 123 040 .ASCII /US /
2129 012764 106 101 111 .ASCII /FAI/
2130 012767 114 125 122 .ASCII /LUR/
2131 012772 105 040 115 .ASCII /E M/

```

Line No.	Code	Label	Value 1	Value 2	Value 3	Value 4	Value 5	Value 6
2133								
2134								
2135								
2136	012775	117	104	125				
2137	013000	114	105	040				
2138	013003	067	063	066				
2139	013006	062	040	000				
2140	013011	000						
2141	013012	101	122	122	P.AHG:			
2142	013015	101	131	040				
2143	013020	104	101	124				
2144	013023	101	040	106				
2145	013026	101	111	114				
2146	013031	125	122	105				
2147	013034	040	115	117				
2148	013037	104	125	114				
2149	013042	105	040	067				
2150	013045	063	066	063				
2151	013050	040	000					
2152	013052	115	105	115	P.AHH:			
2153	013055	117	122	131				
2154	013060	040	101	122				
2155	013063	122	101	131				
2156	013066	040	106	101				
2157	013071	111	114	125				
2158	013074	122	105	040				
2159	013077	115	117	104				
2160	013102	125	114	105				
2161	013105	040	067	063				
2162	013110	065	067	000				
2163	013113	000						
2164	013114	111	116	124	P.AHI:			
2165	013117	105	122	115				
2166	013122	105	104	111				
2167	013125	101	124	105				
2168	013130	040	104	111				
2169	013133	101	107	116				
2170	013136	117	123	124				
2171	013141	111	103	040				
2172	013144	115	105	123				
2173	013147	123	101	107				
2174	013152	105	000					
2175	013154	104	101	124	P.AHJ:			
2176	013157	101	040	114				
2177	013162	101	124	105				
2178	013165	040	105	122				
2179	013170	122	117	122				
2180	013173	040	104	125				
2181	013176	122	111	116				
2182	013201	107	040	124				
2183	013204	122	101	116				
2184	013207	123	106	105				
2185	013212	122	000					
2186	013214	123	103	040	P.AHK:			
2187	013217	102	111	124				

ROUTINE DECLARATION SECTION

:ML4AD
:

P.AHG:

P.AHH:

P.AHI:

P.AHJ:

P.AHK:

```

.ASCII /ODU/
.ASCII /LE /
.ASCII /736/
.ASCII /2 /<00>
.ASCII <00>
.ASCII /ARR/
.ASCII /AY /
.ASCII /DAT/
.ASCII /A F/
.ASCII /AIL/
.ASCII /URE/
.ASCII / MO/
.ASCII /DUL/
.ASCII /E 7/
.ASCII /363/
.ASCII / /<00>
.ASCII /MEM/
.ASCII /ORY/
.ASCII / AR/
.ASCII /RAY/
.ASCII / FA/
.ASCII /ILU/
.ASCII /RE /
.ASCII /MOD/
.ASCII /ULE/
.ASCII / 73/
.ASCII /57/<00>
.ASCII <00>
.ASCII /INT/
.ASCII /ERM/
.ASCII /EDI/
.ASCII /ATE/
.ASCII / DI/
.ASCII /AGN/
.ASCII /OST/
.ASCII /IC /
.ASCII /MES/
.ASCII /SAG/
.ASCII /E/<00>
.ASCII /DAT/
.ASCII /A L/
.ASCII /ATE/
.ASCII / ER/
.ASCII /ROR/
.ASCII / DU/
.ASCII /RIN/
.ASCII /G T/
.ASCII /RAN/
.ASCII /SFE/
.ASCII /R/<00>
.ASCII /SC /
.ASCII /BIT/

```

```

2189
2190
2191
2192 013222 040 123 105
2193 013225 124 040 104
2194 013230 125 122 111
2195 013233 116 107 040
2196 013236 124 122 101
2197 013241 116 123 106
2198 013244 105 122 000
2199 013247 000
2200 013250 124 122 117 P.AHL:
2201 013253 125 102 114
2202 013256 105 040 123
2203 013261 110 117 117
2204 013264 124 040 114
2205 013267 117 117 120
2206 013272 040 105 122
2207 013275 122 117 122
2208 013300 123 000
2209 013302 122 110 040 P.AHM:
2210 013305 103 117 116
2211 013310 124 122 117
2212 013313 114 114 105
2213 013316 122 040 105
2214 013321 122 122 117
2215 013324 122 123 000
2216 013327 000
2217 013330 104 122 111 P.AHN:
2218 013333 126 105 040
2219 013336 110 125 116
2220 013341 107 040 101
2221 013344 106 124 105
2222 013347 122 040 115
2223 013352 101 123 123
2224 013355 040 102 125
2225 013360 123 040 124
2226 013363 122 101 116
2227 013366 123 106 105
2228 013371 122 000 000
2229
2230
2231
2232 013374 NIB.SAVE:
2233 013374 .BLKW 3
2234 013402 HW.OR.TBL:
2235 013402 .BLKW 177
2236 014000 PTBL.PTR:
2237 014000 .BLKW 1
2238 014002 OP.NUM.ARR:
2239 014002 .BLKW 1
2240 014004 ARR.INC: .BLKW 1
2241 014006 GOOD.BLK:
2242 014006 .BLKW 1

```

ROUTINE DECLARATION SECTION

:ML4AD
:

P.AHL:

P.AHM:

P.AHN:

NIB.SAVE:

HW.OR.TBL:

PTBL.PTR:

OP.NUM.ARR:

ARR.INC: .BLKW

GOOD.BLK:

```

.ASCII / SE/
.ASCII /T D/
.ASCII /URI/
.ASCII /NG /
.ASCII /TRA/
.ASCII /NSF/
.ASCII /ER/<00>
.ASCII <00>
.ASCII /TRO/
.ASCII /UBL/
.ASCII /E S/
.ASCII /HOO/
.ASCII /T L/
.ASCII /OOP/
.ASCII / ER/
.ASCII /ROR/
.ASCII /S/<00>
.ASCII /RH /
.ASCII /CON/
.ASCII /TRO/
.ASCII /LLE/
.ASCII /R E/
.ASCII /RRO/
.ASCII /RS/<00>
.ASCII <00>
.ASCII /DRI/
.ASCII /VE /
.ASCII /HUN/
.ASCII /G A/
.ASCII /FTE/
.ASCII /R M/
.ASCII /ASS/
.ASCII / BU/
.ASCII /S T/
.ASCII /RAN/
.ASCII /SFE/
.ASCII /R/<00><00>

```

```

3
177
1
1
1
1
1

```



```
2356      :ML4AD
2357      :
2358
2359 015526      000      .BYTE 0
2360 015527      000      .BYTE 0
2361 015530 000020      .WORD 20
2362 015532 177400      .WORD -400
2363 015534      000      .BYTE 0
2364 015535      000      .BYTE 0
2365 015536 000110      .WORD 110
2366 015540 177666      .WORD -112
2367 015542 000001      .WORD 1
2368 015544      000      .BYTE 0
2369 015545      000      .BYTE 0
2370 015546      000      .BYTE 0
2371 015547      000      .BYTE 0
2372 015550      000      .BYTE 0
2373 015551      000      .BYTE 0
2374 015552      000      .BYTE 0
2375 015553      000      .BYTE 0
2376 015554      000      .BYTE 0
2377 015555      000      .BYTE 0
2378 015556      000      .BYTE 0
2379 015557      000      .BYTE 0
2380 015560 140300      .WORD -37500
2381 015562      000      .BYTE 0
2382 015563      000      .BYTE 0
2383 015564      000      .BYTE 0
2384 015565      000      .BYTE 0
2385 015566      000      .BYTE 0
2386 015567      000      .BYTE 0
2387 015570 100300      .WORD -77500
2388 015572      000      .BYTE 0
2389 015573      000      .BYTE 0
2390 015574      000      .BYTE 0
2391 015575      000      .BYTE 0
2392 015576      000      .BYTE 0
2393 015577      000      .BYTE 0
2394 015600      000      .BYTE 0
2395 015601      000      .BYTE 0
2396 015602      000      .BYTE 0
2397 015603      000      .BYTE 0
2398 015604      000      .BYTE 0
2399 015605      000      .BYTE 0
2400 015606      000      .BYTE 0
2401 015607      000      .BYTE 0
2402 015610      000      .BYTE 0
2403 015611      000      .BYTE 0
2404 015612      000      .BYTE 0
2405 015613      000      .BYTE 0
2406 015614      000      .BYTE 0
2407 015615      000      .BYTE 0
2408 015616      000      .BYTE 0
2409 015617      000      .BYTE 0
2410 015620 010000      .WORD 10000
```



```
2412  
2413  
2414  
2415 015622 000  
2416 015623 000  
2417 015624 000  
2418 015625 000  
2419 015626 000  
2420 015627 000  
2421 015630 000  
2422 015631 000  
2423 015632 000  
2424 015633 000  
2425 015634 000  
2426 015635 000  
2427 015636 000  
2428 015637 000  
2429 015640 000  
2430 015641 000  
2431 015642 000  
2432 015643 000  
2433 015644 000  
2434 015645 000  
2435 015646 000  
2436 015647 000  
2437 015650 000  
2438 015651 000  
2439 015652 000  
2440 015653 000  
2441 015654 000  
2442 015655 000  
2443 015656 000  
2444 015657 000  
2445 015660 000  
2446 015661 000  
2447 015662 000000  
2448 015664 001001  
2449 015666 004004  
2450 015670 020020  
2451 015672 003003  
2452 015674 014014  
2453 015676 021460  
2454 015700 005005  
2455 015702 024024  
2456 015704 023023  
2457 015706 017017  
2458 015710 035474  
2459 015712 024465  
2460 015714 021021  
2461 015716 007007  
2462 015720 034034  
2463 015722 022463  
2464 015724 011011  
2465 015726 005444  
2466 015730 026026
```

```
:ML4AD  
:
```

ROUTINE DECLARATION SECTION

```
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.BYTE 0  
.WORD 0  
REM.TBL : .WORD 1001  
.WORD 4004  
.WORD 20020  
.WORD 3003  
.WORD 14014  
.WORD 21460  
.WORD 5005  
.WORD 24024  
.WORD 23023  
.WORD 17017  
.WORD 35474  
.WORD 24465  
.WORD 21021  
.WORD 7007  
.WORD 34034  
.WORD 22463  
.WORD 11011  
.WORD 5444  
.WORD 26026
```

```
2468  
2469  
2470  
2471 015732 033033  
2472 015734 016457  
2473 015736 033472  
2474 015740 014455  
2475 015742 023462  
2476 015744 015015  
2477 015746 025464  
2478 015750 025025  
2479 015752 027027  
2480 015754 037037  
2481 015756 036477  
2482 015760 030. 1  
2483 015762 041  
2484  
2485 015764 064477  
2486 015766 007700  
2487 015770 000  
2488 015771 000  
2489 015772 064477  
2490 015774 000077  
2491 015776 000  
2492 015777 000  
2493 016000 100577  
2494 016002 017700  
2495 016004 077  
2496 016005 000  
2497 016006 100577  
2498 016010 000000  
2499 016012 077  
2500 016013 000  
2501 016014 100577  
2502 016016 010077  
2503 016020 077  
2504  
2505 016022  
2506 016024  
2507 016026  
2508 016030  
2509 016032  
2510  
2511  
2512  
2513  
2514  
2515  
2516 100000  
2517 040000  
2518 020000  
2519 010000  
2520 004000  
2521 002000  
2522 001000
```

:ML4AD
:
ROUTINE DECLARATION SECTION

DT.1:

RH.ADD: .BLKW 1
RH.TYP: .BLKW 1
RH.VEC: .BLKW 1
ML.LUN: .BLKW 1
ML.DUT: .BLKW 1

.WORD 33033
.WORD 16457
.WORD 33472
.WORD 14455
.WORD 23462
.WORD 15015
.WORD 25464
.WORD 25025
.WORD 27027
.WORD 37037
.WORD 36477
.WORD 30471
.BYTE 41
.EVEN
.WORD 64477
.WORD 7700
.BYTE 0
.BYTE 0
.WORD 64477
.WORD 77
.BYTE 0
.BYTE 0
.WORD -77201
.WORD 17700
.BYTE 77
.BYTE 0
.WORD -77201
.WORD 0
.BYTE 77
.BYTE 0
.WORD -77201
.WORD 10077
.BYTE 77
.EVEN
.GLOBL REGDMP, ONEPAS, INTERVEN, PRSN
.GLOBL ERRTHR

BIT15== -100000
BIT14== 40000
BIT13== 20000
BIT12== 10000
BIT11== 4000
BIT10== 2000
BIT09== 1000

```
2524      ;ML4AD
2525      ;
2526      ;
2527      000400      BIT08==      400
2528      000200      BIT07==      200
2529      000100      BIT06==      100
2530      000040      BIT05==      40
2531      000020      BIT04==      20
2532      000010      BIT03==      10
2533      000004      BIT02==      4
2534      000002      BIT01==      2
2535      000001      BIT00==      1
2536      001000      BIT9==      1000
2537      000400      BIT8==      400
2538      000200      BIT7==      200
2539      000100      BIT6==      100
2540      000040      BIT5==      40
2541      000020      BIT4==      20
2542      000010      BIT3==      10
2543      000004      BIT2==      4
2544      000002      BIT1==      2
2545      000001      BIT0==      1
2546      000040      EF.START==      40
2547      000037      EF.RESTART==      37
2548      000036      EF.CONTINUE==      36
2549      000035      EF.NEW==      35
2550      000034      EF.PWR==      34
2551      000340      PRI07==      340
2552      000300      PRI06==      300
2553      000240      PRI05==      240
2554      000200      PRI04==      200
2555      000140      PRI03==      140
2556      000100      PRI02==      100
2557      000040      PRI01==      40
2558      000000      PRI00==      0
2559      000004      EVL==      4
2560      000010      LOT==      10
2561      000020      ADR==      20
2562      000040      IDU==      40
2563      000100      ISR==      100
2564      000200      UAM==      200
2565      000400      BOE==      400
2566      001000      PNT==      1000
2567      002000      PRI==      2000
2568      004000      IXE==      4000
2569      010000      IBE==      10000
2570      020000      IER==      20000
2571      040000      LOE==      40000
2572      100000      HOE==      -100000
2573      004716      T.21=      P.AAA
2574      005012      T.61=      P.AAB
2575      005112      PWR.OFF=    P.AAC
2576      005202      PWR.ON=     P.AAD
2577      005272      UNS.ERR=    P.AAE
2578      005340      VV.NOT.SET= P.AAF
```

```
2580      :ML4AD
2581      :
2582      :
2583      005414  VV.CLEAR=      P.AAG
2584      005470  NO.INIT=       P.AAH
2585      005570  MB.DIS.ERR=    P.AAI
2586      005652  WC.ERR=       P.AAJ
2587      005722  W.ERR=        P.AAK
2588      005764  VV.SET=       P.AAL
2589      006060  PUP.BB=       P.AAM
2590      006150  BB.VV.ERR=    P.AAN
2591      006242  BB.INIT.ERR=  P.AAO
2592      006334  BB.BB.ERR=    P.AAP
2593      006442  FMT.1=       P.AAQ
2594      006506  FMT.2=       P.AAR
2595      006574  FMT.3=       P.AAS
2596      006620  FMT.4=       P.AAT
2597      006650  FMT.5=       P.AAU
2598      006752  FMT.6=       P.AAV
2599      007002  FMT.7=       P.AAW
2600      007030  FMT.8=       P.AAX
2601      007064  FMT.9=       P.AAY
2602      007116  FMT.10=      P.AAZ
2603      007170  FMT.11=     P.ABA
2604      007224  FMT.12=     P.ABB
2605      007254  FMT.13=     P.ABC
2606      007330  FMT.14=     P.ABD
2607      007370  FMT.15=     P.ABE
2608      007436  FMT.16=     P.ABF
2609      007526  FMT.17=     P.ABG
2610      007562  FMT.18=     P.ABH
2611      007632  FMT.19=     P.ABI
2612      007702  FMT.20=     P.ABJ
2613      007754  FMT.21=     P.ABK
2614      010026  FMT.22=     P.ABL
2615      010072  FMT.23=     P.ABM
2616      010142  FMT.24=     P.ABN
2617      010166  FMT.25=     P.ABO
2618      010206  FMT.26=     P.ABP
2619      010240  ONE.FMT=    P.ABQ
2620      010246  TWO.FMT=    P.ABR
2621      010256  THR.FMT=    P.ABS
2622      010270  FOR.FMT=    P.ABT
2623      010304  FIV.FMT=    P.ABU
2624      010322  SIX.FMT=    P.ABV
2625      010342  SEV.FMT=    P.ABW
2626      010364  EIG.FMT=    P.ABX
2627      010410  NIN.FMT=    P.ABY
2628      010436  TEN.FMT=    P.ABZ
2629      010466  ELV.FMT=    P.ACA
2630      010520  WRD.1=     P.ACB
2631      010524  WRD.2=     P.ACC
2632      010536  WRD.3=     P.ACD
2633      010544  WRD.4=     P.ACE
2634      010552  WRD.5=     P.ACF
```

ROUTINE DECLARATION SECTION

ROUTINE DECLARATION SECTION

2636		:ML4AD	
2637		:	
2638		:	
2639	010560	WRD.6=	P.ACG
2640	010566	WRD.7=	P.ACH
2641	010602	WRD.8=	P.ACI
2642	010616	WRD.9=	P.ACJ
2643	010630	WRD.10=	P.ACK
2644	010640	WRD.11=	P.ACL
2645	010650	WRD.12=	P.ACM
2646	010660	WRD.13=	P.ACN
2647	010664	WRD.14=	P.ACO
2648	010676	WRD.15=	P.ACP
2649	010704	WRD.16=	P.ACQ
2650	010712	WRD.17=	P.ACR
2651	010724	WRD.18=	P.ACS
2652	010730	WRD.19=	P.ACT
2653	010736	WRD.20=	P.ACU
2654	010744	WRD.21=	P.ACV
2655	010752	WRD.22=	P.ACW
2656	010766	WRD.23=	P.ACX
2657	010774	WRD.24=	P.ACY
2658	011002	WRD.25=	P.ACZ
2659	011016	WRD.26=	P.ADA
2660	011024	WRD.27=	P.ADB
2661	011032	WRD.29=	P.ADC
2662	011046	WRD.30=	P.ADD
2663	011054	WRD.31=	P.ADE
2664	011070	WRD.32=	P.ADF
2665	011076	WRD.33=	P.ADG
2666	011102	WRD.34=	P.ADH
2667	011110	WRD.35=	P.ADI
2668	011116	WRD.36=	P.ADJ
2669	011122	WRD.37=	P.ADK
2670	011132	WRD.38=	P.ADL
2671	011140	WRD.39=	P.ADM
2672	011150	WRD.40=	P.ADN
2673	011154	WRD.41=	P.ADO
2674	011170	WRD.42=	P.ADP
2675	011176	WRD.43=	P.ADQ
2676	011206	WRD.44=	P.ADR
2677	011214	WRD.45=	P.ADS
2678	011226	WRD.46=	P.ADT
2679	011240	WRD.47=	P.ADU
2680	011250	WRD.48=	P.ADV
2681	011260	WRD.49=	P.ADW
2682	011270	WRD.50=	P.ADX
2683	011276	WRD.51=	P.ADY
2684	011310	WRD.52=	P.ADZ
2685	011316	WRD.53=	P.AEA
2686	011326	WRD.54=	P.AEB
2687	011336	WRD.55=	P.AEC
2688	011344	WRD.56=	P.AED
2689	011352	WRD.57=	P.AEE
2690	011362	WRD.58=	P.AEF

ROUTINE DECLARATION SECTION

2692		:ML4AD	
2693		:	
2694			
2695	011374	WRD.59=	P.AEG
2696	011400	WRD.60=	P.AEH
2697	011416	WRD.61=	P.AEI
2698	011432	WRD.62=	P.AEJ
2699	011440	WRD.63=	P.AEK
2700	011446	WRD.64=	P.AEL
2701	011454	WRD.65=	P.AEM
2702	011462	WRD.67=	P.AEN
2703	011470	WRD.68=	P.AEO
2704	011474	WRD.69=	P.AEP
2705	011502	WRD.70=	P.AEQ
2706	011514	WRD.71=	P.AER
2707	011524	WRD.72=	P.AES
2708	011532	WRD.73=	P.AET
2709	011540	WRD.74=	P.AEU
2710	011552	WRD.75=	P.AEV
2711	011562	WRD.76=	P.AEW
2712	011570	WRD.77=	P.AEX
2713	011610	PHR.1=	P.AEY
2714	011626	PHR.2=	P.AEZ
2715	011644	PHR.3=	P.AFA
2716	011676	PHR.4=	P.AFB
2717	011714	PHR.5=	P.AFC
2718	011726	PHR.6=	P.AFD
2719	011740	PHR.7=	P.AFE
2720	011762	PHR.8=	P.AFF
2721	011774	PHR.9=	P.AFG
2722	012006	PHR.10=	P.AFH
2723	012020	PHR.11=	P.AFI
2724	012036	PHR.12=	P.AFJ
2725	012060	PHR.13=	P.AFK
2726	012104	PHR.14=	P.AFL
2727	012136	PHR.15=	P.AFM
2728	012150	FNC.1=	P.AFN
2729	012164	FNC.2=	P.AFO
2730	012172	FNC.3=	P.AFP
2731	012200	FNC.4=	P.AFQ
2732	012216	FNC.5=	P.AFR
2733	012226	FNC.6=	P.AFS
2734	012234	FNC.7=	P.AFT
2735	012244	FNC.8=	P.AFU
2736	012260	FNC.9=	P.AFV
2737	012272	FNC.10=	P.AFW
2738	012302	FNC.11=	P.AFX
2739	012322	FNC.12=	P.AFY
2740	012334	FNC.13=	P.AFZ
2741	012344	FNC.14=	P.AGA
2742	012360	FNC.15=	P.AGB
2743	012372	FNC.16=	P.AGC
2744	012404	FNC.17=	P.AGD
2745	012414	FNC.18=	P.AGE
2746	012426	FNC.19=	P.AGF

```

2748      :ML4AD
2749      :
2750      :
2751      012440      FNC.21=      P.AGG
2752      012452      FNC.22=      P.AGH
2753      012464      FNC.23=      P.AGI
2754      012500      REG.1=       P.AGJ
2755      012510      REG.2=       P.AGK
2756      012516      REG.3=       P.AGL
2757      012524      REG.4=       P.AGM
2758      012532      REG.5=       P.AGN
2759      012540      REG.6=       P.AGO
2760      012546      REG.7=       P.AGP
2761      012554      REG.8=       P.AGQ
2762      012562      REG.9=       P.AGR
2763      012570      REG.10=      P.AGS
2764      012576      REG.11=      P.AGT
2765      012604      REG.12=      P.AGU
2766      012612      REG.13=      P.AGV
2767      012620      REG.14=      P.AGW
2768      012626      REG.15=      P.AGX
2769      012634      REG.16=      P.AGY
2770      012642      REG.17=      P.AGZ
2771      012652      REG.18=      P.AHA
2772      012660      REG.19=      P.AHB
2773      012666      REG.20=      P.AHC
2774      012676      REG.21=      P.AHD
2775      012706      ASYNC=       P.AHE
2776      012750      SYNC=       P.AHF
2777      013012      ARR.DAT=    P.AHG
2778      013052      MEM.ARR=    P.AHH
2779      013114      INTER=     P.AHI
2780      013154      DATA.LATE= P.AHJ
2781      013214      SC.SET=     P.AHK
2782      013250      TRBLE.LOOP= P.AHL
2783      013302      RH.ERROR=   P.AHM
2784      013330      TIME.OUT=   P.AHN
2785      013374      D1.TEMP=    NIB.SAVE
2786      013376      D2.TEMP=    NIB.SAVE+2
2787      013400      E2.TEMP=    NIB.SAVE+4
2788
2789
2790
2791

```

.SBTTL LOAD.STACK ROUTINE DECLARATION SECTION

```

2795 016034      004167      166514
2796 016034      016601      000012
2797 016040      012702      015022
2798 016044      060102
2799 016050      016601      000014
2800 016052      016600      000012
2801 016056

```

```

LOAD.STACK:
JSR      R1,$SAVE3
MOV      12(SP),R1
MOV      #STK.OFF,R2
ADD      R1,R2
MOV      14(SP),R1
MOV      12(SP),R0
:
: NIB.PTR,*
:
: STK.PTR,*
: NIB.PTR,*

```

2267
2322

2318

2859									
2860			:ML4AD						
2861			:						
2862	016250	005003							
2863	016252	151203	9\$:	CLR	R3	:			
2864	016254	010100		BISB	(R2),R3				2346
2865	016256	160300		MOV	R1,R0				
2866	016260	016703	175112	SUB	R3,R0				
2867	016264	000421		MOV	NIB.SAVE+2,R3				
2868	016266	005003		BR	13\$				
2869	016270	151203	10\$:	CLR	R3	:			
2870	016272	010100		BISB	(R2),R3				2350
2871	016274	160300		MOV	R1,R0				
2872	016276	016703	175074	SUB	R3,R0				
2873	016302	006203		MOV	NIB.SAVE+2,R3				
2874	016304	006203	11\$:	ASR	R3				
2875	016306	006203		ASR	R3				
2876	016310	006203		ASR	R3				
2877	016312	000406		ASR	R3				
2878	016314	005003		BR	13\$				
2879	016316	151203	12\$:	CLR	R3	:			
2880	016320	010100		BISB	(R2),R3				2354
2881	016322	160300		MOV	R1,R0				
2882	016324	016703	175050	SUB	R3,R0				
2883	016330	000303		MOV	NIB.SAVE+4,R3				
2884	016332	042703	177760	13\$:	SWAB	R3			
2885	016336	105060	015034	14\$:	BIC	#177760,R3			
2886	016342	150360	015034		CLRB	STACK(R0)			
2887	016346	000207		BISB	R3,STACK(R0)				
2888	016350	005003		RTS	PC	:			2318
2889	016352	151203		15\$:	CLR	R3	:		2358
2890	016354	160301		BISB	(R2),R3				
2891	016356	016703	175016	SUB	R3,R1				
2892	016362	006203		MOV	NIB.SAVE+4,R3				
2893	016364	006203		ASR	R3				
2894	016366	006203		ASR	R3				
2895	016370	006203		ASR	R3				
2896	016372	000303		ASR	R3				
2897	016374	042703	177770	SWAB	R3				
2898	016400	105061	015034	BIC	#177770,R3				
2899	016404	150361	015034		CLRB	STACK(R1)			
2900	016410	000207		BISB	R3,STACK(R1)				
2901				RTS	PC	:			2267

: Routine Size: 119 words
: Maximum stack depth per invocation: 4 words

2902
2903
2908
2909

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (5)

2911 :ML4AD
 2912 :
 2913 :
 2914 :
 2915 :
 2916 :
 2917 :
 2918 :
 2919 :
 2920 :
 2921 :
 2922 :
 2923 :
 2924 :
 2925 :
 2926 :
 2927 :
 2928 :
 2929 :
 2930 :
 2931 :
 2932 :
 2933 :
 2934 :
 2935 :
 2936 :
 2937 :
 2941 :
 2942 :
 2946 016412
 2947 016412 152777
 2948 016420 016777
 2949 016426 012777
 2950 016434 012777
 2951 016442 000207
 2952 :
 2953 :
 2954 :
 2959 :
 2960 :

ROUTINE DECLARATION SECTION

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 COMPLEMENT 256
    
```

.SBTTL DAT.DM.XFER ROUTINE DECLARATION SECTION

```

DAT.DM.XFER:
    BISB #10,AML.REG+120
    MOV GOOD.BLK,AML.REG+30
    MOV #IO.BUF,AML.REG+20
    MOV #-400,AML.REG+10
    RTS PC
    
```

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

2382
 2383
 2384
 2385
 2363

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (6)

```

2962 :ML4AD
2963 :
2964 :
2965 : 2387 routine STRIPPER (WRD_CNT, NIB_CNT) : novalue =
2966 : 2388     begin
2967 : 2389
2968 : 2390 !++
2969 : 2391 !FUNCTIONAL DESCRIPTION:
2970 : 2392 !STRIPPER RUNS IN DATA DIAG MODE AND STRIPS OUT
2971 : 2393 !A VARIABLE NUMBER OF GOOD NIBBLES FROM THE
2972 : 2394 !GOOD BLOCK AND STORES THEM INTO A CONTIGIOUS
2973 : 2395 !STACK FOR SEQUENTIAL ACCESSING
2974 : 2396
2975 : 2397 !FORMAL PARAMETERS:
2976 : 2398 !WRD_CNT
2977 : 2399 !TELS STRIPPER HOW MANY WORDS IN THE GOOD
2978 : 2400 !BLOCK TO READ OUT
2979 : 2401
2980 : 2402 !NIB_CNT
2981 : 2403 !TELS STRIPPER HOW MANY NIBBLES TO
2982 : 2404 !STRIP OUT OF EACH WORD.
2983 : 2405
2984 : 2406 !IMPLICIT INPUTS:
2985 : 2407 !--
2986 : 2408
2987 : 2409     local
2988 : 2410         STK_PTR;                !STACK POINTER
2989 : 2411
2990 : 2412     incr CNT from 0 to .NIB_CNT do    !CLEAR OUT THE STACK OFFSETS
2991 : 2413         STK_OFF [.CNT] = ZEROES;
2992 : 2414
2993 : 2415     CLR_MBUS;
2994 : 2416     STK_PTR = -1;                !RESET THE STACK POINTER
2995 : 2417     DAT_DM_XFER ();            !SET UP A DATA XFER AT THE GOOD BLOCK
2996 : 2418     MLC51 = read;              !DO A READ XFER
2997 : 2419     DELAY (ONE_US);            !ALLOW PROM DATA TO GET INTO THE MLPD REG
2998 : 2420
2999 : 2421     incr CNT from 0 to .WRD_CNT do    !LOAD THE STACK WITH ALL GOOD NIBBLE DATA
3000 : 2422     begin
3001 : 2423         PD_TEMP = .MLPD;        !GET THE PROM DATA
3002 : 2424         DAT_CLK = ONE;         !CLOCK OUT THE DATA WORD
3003 : 2425         DELAY (ONE_US);
3004 : 2426         RD_LNG_WRD;           !READ DATA DIAG REGS INTO NIBBLE SAVE
3005 : 2427
3006 : 2428         incr NIB_PTR from 0 to .NIB_CNT do    !STRIP OUT X NUMBER OF NIBBLES
3007 : 2429         begin
3008 : 2430             STK_PTR = .STK_PTR + 1;        !INCREMENT THE STACK POINTER
3009 : 2431
3010 : 2432             if .PD_TEMP [.NIB_PTR] IS_SET    !SEE IF THIS A GOOD NIBBLE
3011 : 2433             then
3012 : 2434                 STK_OFF [.NIB_PTR] = (.STK_OFF [.NIB_PTR]) + (.NIB_CNT + 1)
3013 : 2435                 !THEN INCREMENT IT'S STACK OFFSET
3014 : 2436             else
3015 : 2437                 LOAD_STACK (.STK_PTR, .NIB_PTR);    !ELSE LOAD THE NIBBLE IN THE STACK
3016 : 2438
    
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (6)

```

3018 :ML4AD
3019 :
3020 :
3021 : 2439          end:
3022 : 2440
3023 : 2441          end:
3024 : 2442
3025 : 2443          CLR MBUS;
3026 : 2444          end:
    
```

```

3030 :
3031 :
3032 :
3033 :
3034 :
3035 :
3036 :
3037 :
3038 :
3039 :
3040 :
3041 :
3042 :
3043 :
3044 :
3045 :
3046 :
3047 :
3048 :
3049 :
3050 :
3051 :
3052 :
3053 :
3054 :
3055 :
3056 :
3057 :
3058 :
3059 :
3060 :
3061 :
3062 :
3063 :
3064 :
3065 :
3066 :
3067 :
3068 :
3069 :
3070 :
3071 :
3072 :
    
```

.GLOBL LSDLY

.SBTTL STRIPPER ROUTINE DECLARATION SECTION

```

3038 016444          STRIPPER:
3039 016444 004167 166142      JSR    R1,$SAVE5
3040 016450 005746          TST    -(SP)
3041 016452 016604 000020      MOV    20(SP),R4
3042 016456 005001          CLR    R1
3043 016460 000403          BR     2$
3044 016462 105061 015022      1$:   CLRB  STK.OFF(R1)
3045 016466 005201          INC    R1
3046 016470 020104          2$:   CMP    R1,R4
3047 016472 003773          BLE   1$
3048 016474 152777 000040 176742  BISB  #40,2ML.REG+40
3049 016502 016705 177324      MOV    ML,DUT,R5
3050 016506 042705 177770      BIC   #177770,R5
3051 016512 142777 000007 176724  BICB  #7,2ML.REG+40
3052 016520 150577 176720      BISB  R5,2ML.REG+40
3053 016524 012705 177777      MOV    #-1,R5
3054 016530 004767 177656          JSR   PC,DAT.DM.XFER
3055 016534 012777 000071 176642  MOV    #71,2ML.REG
3056 016542 012701 000001          MOV   #1,R1
3057 016546 001410          3$:   BEQ   6$
3058 016550 016702 163342      MOV   LSDLY,R2
3059 016554 001403          BEQ   5$
3060 016556 005016          4$:   CLR  (SP)
3061 016560 005302          DEC  R2
3062 016562 001375          BNE  4$
3063 016564 005301          5$:   DEC  R1
3064 016566 000767          BR   3$
3065 016570 005003          6$:   CLR  R3
3066 016572 000502          BR   16$
3067 016574 007767 177034 176540 7$:   MOV   2ML.REG+230,PD.TEMP
3068 016602 152777 000020 176714  BISB  #20,2ML.REG+120
3069 016610 012701 000001          MOV   #1,R1
3070 016614 001410          8$:   BEQ   11$
3071 016616 016702 163274      MOV   LSDLY,R2
3072 016622 001403          BEQ   10$
    
```

2387
 2412
 2413
 2412
 2413
 2416
 2417
 2418
 2419
 2421
 2423
 2424
 2425

3130
3131
3132
3137
3138

:ML4AD
:

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (7)

```

3140 :ML4AD
3141 :
3142 :
3143 : 2445 routine CAL_CRC : novalue =
3144 : 2446   begin
3145 : 2447
3146 : 2448
3147 : 2449
3148 : 2450
3149 : 2451
3150 : 2452
3151 : 2453
3152 : 2454
3153 : 2455
3154 : 2456
3155 : 2457
3156 : 2458
3157 : 2459
3158 : 2460
3159 : 2461
3160 : 2462
3161 : 2463
3162 : 2464
3163 : 2465
3164 : 2466
3165 : 2467
3166 : 2468
3167 : 2469
3168 : 2470
3169 : 2471
3170 : 2472
3171 : 2473
3172 : 2474
3173 : 2475
3174 : 2476
3175 : 2477
3176 : 2478
3177 : 2479
3178 : 2480
3179 : 2481
3180 : 2482
3181 : 2483
3182 : 2484
3183 : 2485
3184 : 2486
3185 : 2487
3186 : 2488
3187 : 2489
3188 : 2490
3189 : 2491
3190 : 2492
3191 : 2493
3192 : 2494
3193 : 2495
3194 : 2496

ROUTINE DECLARATION SECTION

++
FUNCTIONAL DESCRIPTION:
  CALCULATE THE CRC CODE FOR THE GOOD BLOCKS
  FIRST CRC GROUP (60 NIBBLES) BY STRIPPING
  OUT GOOD NIBBLES AND MODULO 2 ADDING EACH
  BITS WEIGHT RESULTING IN A_CAL,B_CAL,P_CAL.

IMPLICIT INPUTS:
  P_CAL,A_CAL,B_CAL
  GLOBAL STORAGE LOCATION WHERE CALCULATED
  CRC CODES ARE STORED AND EXAMINED FROM

  REM TBL
  TABLE CONTAINING PRECALCULATED BIT POSITION
  WEIGHTS USED IN THE MODULO 2 CALCULATION OF THE
  CRC CODE
--

local
  CHANNEL,                !CHANNEL POINTS TO THE BITS IN A WORD
  STK_PTR,                !STACK POINTER
  NIB_SAV : bitvector [4], !NIBBLE SAVE LOCATION
  ALOG,                   !INDEX INTO REMAINDER TABLE
  BLOG;                   !INDEX INTO REMAINDER TABLE

STRIPPER (12, 9);        !STRIP OUT ALL 10 NIBBLE FROM 13 ARRAY WORDS
STK_PTR = -1;           !RESET STACK POINTER

incr PLOG from 0 to 5 do !PLOG POINTS TO ONE CRC GROUP '6 WORDS'
  begin
  CHANNEL = -1;          !RESET CHANNEL FOR EACH WORD

  incr NIB_CNT from 0 to 8 do !READ DATA NIBBLE 0-8 AND CALCULATE CRC
    begin
    STK_PTR = .STK_PTR + 1; !INCREMENT THE STACK POINTER
    NIB_SAV = .stack [.STK_PTR]; !LOAD NIB_SAV WITH A STACK NIBBLE

    incr BIT_TST from 0 to 3 do !TEST THE BIT FOR BEING SET
      begin
      CHANNEL = .CHANNEL + 1; !CHANNEL POINTS TO THIS BIT

      if .NIB_SAV [.BIT_TST] IS_SET !SEE IF THIS BIT IS SET
      then
      begin
      !IF SET THEN MOD 2 ADD THE REMAINDER
      ALOG = .PLOG + .CHANNEL; !CALCULATE THE ALOG
      BLOG = .PLOG + .CHANNEL*2; !CALCULATE THE BLOG

      while .ALOG geq 63 do !REDUCE ALOG UNTIL < 64
        ALOG = .ALOG - 63;

```

```

3196 ;ML4AD
3197 :
3198 :
3199 : 2497
3200 : 2498
3201 : 2499
3202 : 2500
3203 : 2501
3204 : 2502
3205 : 2503
3206 : 2504
3207 : 2505
3208 : 2506
3209 : 2507
3210 : 2508
3211 : 2509
3212 : 2510
3213 : 2511
3214 : 2512
3215 : 2513
3219 :
3220 :
  
```

```

ROUTINE DECLARATION SECTION

while .BLOG geq 63 do          !REDUCE BLOG UNTIL <64
  BLOG = .BLOG - 63;

  P_CAL = (.P_CAL) xor (.REM_TBL [.PLOG]);    !MOD 2 P_CAL WITH REM_TBL
  A_CAL = (.A_CAL) xor (.REM_TBL [.ALOG]);    !MOD 2 A_CAL WITH REM_TBL
  B_CAL = (.B_CAL) xor (.REM_TBL [.BLOG]);    !MOD 2 B_CAL WITH REM_TBL
end;

end;

STK_PTR = .STK_PTR + 1;      !SKIP OVER THE CRC NIB NT THE STACK
end;

end;
  
```

```

3224 017042 004167 165544
3225 017046 162706 000012
3226 017052 012746 000014
3227 017056 012746 000011
3228 017062 004767 177356
3229 017066 012766 177777 000006
3230 017074 005066 000004
3231 017100 012766 177777 000010 1$:
3232 017106 005066 000012
3233 017112 005266 000006 2$:
3234 017116 016605 000006
3235 017122 116566 015034 000014
3236 017130 105066 000015
3237 017134 005003
3238 017136 005266 000010 3$:
3239 017142 010305
3240 017144 006205
3241 017146 006205
3242 017150 006205
3243 017152 012704 000014
3244 017156 060604
3245 017160 060405
3246 017162 010546
3247 017164 010346
3248 017166 042716 177770
3249 017172 012746 000001
3250 017176 005046
  
```

```

.SBTTL CAL.CRC ROUTINE DECLARATION SECTION
CAL.CRC:JSR R1,$SAVE5
SUB #12,SP
MOV #14,-(SP)
MOV #11,-(SP)
JSR PC,STRIPPER
MOV #-1,6(SP)
CLR 4(SP)
MOV #-1,10(SP)
CLR 12(SP)
INC 6(SP)
MOV 6(SP),R5
MOVB STACK(R5),14(SP)
CLRB 15(SP)
CLR R3
INC 10(SP)
MOV R3,R5
ASR R5
ASR R5
ASR R5
MOV #14,R4
ADD SP,R4
ADD R4,R5
MOV R5,-(SP)
MOV R3,-(SP)
BIC #177770,(SP)
MOV #1,-(SP)
CLR -(SP)
  
```

```

: * .STK_PTR
: PLOG
: * CHANNEL
: NIB.CNT
: STK_PTR
: STK_PTR,*
: * NIB.SAV
: NIB.SAV
: BIT.TST
: CHANNEL
: BIT.TST,*
: NIB.SAV,*
: BIT.TST,*
  
```

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Line No.
3252				:ML4AD			
3253				:			
3254					ROUTINE DECLARATION SECTION		
3255	017200	004707	164430		JSR PC,BLSGT2		
3256	017204	062706	000010		ADD #10,SP		
3257	017210	005300			DEC R0		
3258	017212	001066			BNE 7\$		
3259	017214	016601	000004		MOV 4(SP),R1	: PLOG,ALOG	2492
3260	017220	066601	000010		ADD 10(SP),R1	: CHANNEL,ALOG	
3261	017224	016605	000010		MOV 10(SP),R5	: CHANNEL,*	2493
3262	017230	006305			ASL R5		
3263	017232	066605	000004		ADD 4(SP),R5	: PLOG,*	
3264	017236	010502			MOV R5,R2	: *,BLOG	
3265	017240	020127	000077	4\$:	CMP R1,#77	: ALOG,*	2495
3266	017244	002403			BLT 5\$		
3267	017246	162701	000077		SUB #77,R1	: *,ALOG	2496
3268	017252	000772			BR 4\$		2495
3269	017254	020227	000077	5\$:	CMP R2,#77	: BLOG,*	2498
3270	017260	002403			BLT 6\$		
3271	017262	162702	000077		SUB #77,R2	: *,BLOG	2499
3272	017266	000772			BR 5\$		2498
3273	017270	005005		6\$:	CLR R5		2501
3274	017272	016604	000004		MOV 4(SP),R4	: PLOG,*	
3275	017276	156405	015664		BISB REM.TBL(R4),R5		
3276	017302	010546			MOV R5,-(SP)		
3277	017304	046716	176056		BIC P.CAL,(SP)		
3278	017310	040567	176052		BIC R5,P.CAL		
3279	017314	052667	176046		BIS (SP)+,P.CAL		
3280	017320	005005			CLR R5		
3281	017322	156105	015664		BISB REM.TBL(R1),R5	: *(ALOG),*	2502
3282	017326	010504			MOV R5,R4		
3283	017330	046704	176026		BIC A.CAL,R4		
3284	017334	040567	176022		BIC R5,A.CAL		
3285	017340	050467	176016		BIS R4,A.CAL		
3286	017344	005005			CLR R5		
3287	017346	156205	015664		BISB REM.TBL(R2),R5	: *(BLOG),*	2503
3288	017352	010504			MOV R5,R4		
3289	017354	046704	176004		BIC B.CAL,R4		
3290	017360	040567	176000		BIC R5,B.CAL		
3291	017364	050467	175774		BIS R4,B.CAL		
3292	017370	005203		7\$:	INC R3	: BIT.TST	
3293	017372	020327	000003		CMP R3,#3	: BIT.TST,*	2485
3294	017376	003657			BLE 3\$		
3295	017400	005266	000012		INC 12(SP)	: NIB.CNT	2480
3296	017404	026627	000012 000010		CMP 12(SP),#10	: NIB.CNT,*	
3297	017412	003637			BLE 2\$		
3298	017414	005266	000006		INC 6(SP)	: STK.PTR	2510
3299	017420	005266	000004		INC 4(SP)	: PLOG	2476
3300	017424	026627	000004 000005		CMP 4(SP),#5	: PLOG,*	
3301	017432	003622			BLE 1\$		
3302	017434	062706	000016		ADD #16,SP		
3303	017440	000207			RTS PC		2445

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

3308
3309
3310
3315
3316

:ML4AD
:

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (8)

```

3318 :ML4AD
3319 :
3320 :
3321 : 2514 routine ERR_CHK_CRC =
3322 : 2515 begin
3323 : 2516
3324 : 2517
3325 : 2518
3326 : 2519
3327 : 2520
3328 : 2521
3329 : 2522
3330 : 2523
3331 : 2524
3332 : 2525
3333 : 2526
3334 : 2527
3335 : 2528
3336 : 2529
3337 : 2530
3338 : 2531
3339 : 2532
3340 : 2533
3341 : 2534
3342 : 2535
3343 : 2536
3344 : 2537
3345 : 2538
3346 : 2539
3347 : 2540
3348 : 2541
3349 : 2542
3350 : 2543
3351 : 2544
3352 : 2545
3353 : 2546
3354 : 2547
3355 : 2548
3356 : 2549
3357 : 2550
3358 : 2551
3362 :
3363 :
3367 : 017442
3368 : 017442
3369 : 017446
3370 : 017452
3371 : 017456
3372 : 017460
    
```

```

ROUTINE DECLARATION SECTION

routine ERR_CHK_CRC =
begin

++
FUNCTIONAL DESCRIPTION:
'ERROR CHECK CRC' COMPARES HARDWARE GENERATED CRC
CODES TO SOFTWARE CALCULATED CRC CODE FOR ONE
CRC GROUP AND RETURNS A ONE ON DETECTION OF ERRORS

IMPLICIT INPUTS:
P_CAL,A_CAL,B_CAL
GLOBAL STORAGE LOCATIONS WHERE CALCULATED
CRC CODES ARE STORED AND EXAMINED FROM

P_GEN,A_GEN,B_GEN
GLOBAL STROAGE LOCATIONS WHERE HARDWARE GENERATED
CRC CODES ARE STORED AND EXAMINED FROM
--

local
POS,
TEMP;
!VARIABLE FOR FIELD SELECTOR
!TEMPORARY STORAGE LOCATION

POS = -1;
!RESET THE POS

incr STK_PTR from 9 to 59 by 10 do
begin
POS = .POS + 1;
!INCREMENT POS
TEMP = .stack [.STK_PTR];
!READ CRC NIB FROM THE STACK
(P_GEN)<.POS, 1> = .TEMP<0, 1>;
!CONVERT 1ST BIT
(A_GEN)<.POS, 1> = not .TEMP<1, 1>;
!CONVERT 2ND BIT
(B_GEN)<.POS, 1> = not .TEMP<2, 1>;
!CONVERT 3ED BIT
end;

if (.P_GEN neq .P_CAL) or (.A_GEN neq .A_CAL) or (.B_GEN neq .B_CAL) then return ONE else return ZERO;
!COMPARE GENERATED CRC CODE TO CALCULATED CRC CODE

end;
    
```

```

.SBTTL ERR.CHK.CRC ROUTINE DECLARATION SECTION
ERR.CHK.CRC:
JSR R1,$SAVE3
MOV #-1,R1
MOV #11,R2
1$: INC R1
CLR R3
:
: *.POS
: *.STK.PTR
: POS
: TEMP
    
```

2514
 2537
 2539
 2541
 2542

Address	Hex	Hex	Hex	Label	Instruction	Comments	Address
3374				:ML4AD			
3375				:			
3376					ROUTINE DECLARATION SECTION		
3377	017462	156203	015034		BISB STACK(R2),R3	: *(STK.PTR),TEMP	
3378	017466	012746	015374		MOV #P.GEN,-(SP)	:	
3379	017472	010146			MOV R1,-(SP)	: POS,*	2543
3380	017474	012746	000001		MOV #1,-(SP)	:	
3381	017500	010346			MOV R3,-(SP)	: TEMP,*	
3382	017502	042716	177776		BIC #177776,(SP)	:	
3383	017506	004767	164360		JSR PC,BL\$PU2		
3384	017512	012716	015370		MOV #A.GEN,(SP)	:	
3385	017516	010146			MOV R1,-(SP)	: POS,*	2544
3386	017520	012746	000001		MOV #1,-(SP)	:	
3387	017524	005046			CLR -(SP)		
3388	017526	032703	000002		BIT #2,R3	: *,TEMP	
3389	017532	001401			BEQ 2\$		
3390	017534	005216			INC (SP)		
3391	017536	005116		2\$:	COM (SP)		
3392	017540	004767	164326		JSR PC,BL\$PU2		
3393	017544	012716	015372		MOV #B.GEN,(SP)	:	
3394	017550	010146			MOV R1,-(SP)	: POS,*	2545
3395	017552	012746	000001		MOV #1,-(SP)	:	
3396	017556	005046			CLR -(SP)		
3397	017560	032703	000004		BIT #4,R3	: *,TEMP	
3398	017564	001401			BEQ 3\$		
3399	017566	005216			INC (SP)		
3400	017570	005116		3\$:	COM (SP)		
3401	017572	004767	164274		JSR PC,BL\$PU2		
3402	017576	062706	000024		ADD #24,SP	:	
3403	017602	062702	000012		ADD #12,R2	: *,STK.PTR	2540
3404	017606	020227	000073		CMP R2,#73	: STK.PTR,*	2539
3405	017612	003721			BLE 1\$		
3406	017614	026767	175554 175544		CMP P.GEN,P.CAL	:	
3407	017622	001010			BNE 4\$		2548
3408	017624	026767	175540 175530		CMP A.GEN,A.CAL		
3409	017632	001004			BNE 4\$		
3410	017634	026767	175532 175522		CMP B.GEN,B.CAL		
3411	017642	001403			BEQ 5\$		
3412	017644	012701	000001	4\$:	MOV #1,R1		
3413	017650	000401			BR 6\$		
3414	017652	005001		5\$:	CLR R1		
3415	017654	010100		6\$:	MOV R1,R0	:	
3416	017656	000207			RTS PC	:	2515
3417						:	2514
3418							
3419							
3424							
3425							

: Routine Size: 71 words
 : Maximum stack depth per invocation: 14 words

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (9)

3427 :ML4AD
 3428 :
 3429 :
 3430 :
 3431 :
 3432 :
 3433 :
 3434 :
 3435 :
 3436 :
 3437 :
 3438 :
 3439 :
 3440 :
 3441 :
 3442 :
 3443 :
 3444 :
 3445 :
 3446 :
 3447 :
 3448 :
 3449 :
 3450 :
 3451 :
 3452 :
 3453 :
 3454 :
 3455 :
 3456 :
 3457 :
 3458 :
 3459 :
 3460 :
 3461 :
 3462 :
 3463 :
 3464 :
 3465 :
 3466 :
 3470 :
 3471 :

2552
 2553
 2554
 2555
 2556
 2557
 2558
 2559
 2560
 2561
 2562
 2563
 2564
 2565
 2566
 2567
 2568
 2569
 2570
 2571
 2572
 2573
 2574
 2575
 2576
 2577
 2578
 2579
 2580
 2581
 2582
 2583
 2584
 2585
 2586
 2587
 2588

ROUTINE DECLARATION SECTION

routine FIND_COMP_BIT (BITS_XFERED) =
 begin

++
 FUNCTIONAL DESCRIPTION:
 'FIND COMPLIMENT' BIT IS CALLED BY THE
 SYNDROME DECODE TESTS AND INDEXES INTO
 THE IO_BUF LOOKING FOR COMPLIMENTED
 BITS. A COMPLIMENTED BIT IS BY DEF A ONE
 A ONE IS RETURNED IF THE INDEXED BIT IS
 NOT SET

FORMAL PARAMETERS:
 BITS_XFERED
 REPRESENTS THE BIT BEING TESTED
 IN THE DRIVE AND FROM THIS THE
 WORD AND BIT INDEX INTO THE
 IO_BUF CAN BE CALCULATED

IMPLICIT INPUTS:
 --

```

local
  COMP_WRD,           !WORD WHERE COMPLIMENT BIT IS LOCATED
  COMP_BIT,          !BIT THAT IS COMPLIMENTED
  BUF_SAV : bitvector [16]; !STORES THE WORD IN WHICH THE BIT IS COMPLIMENTED

  COMP_WRD = .BITS_XFERED/16; !CALCULATE THE COMP WORD
  COMP_BIT = .BITS_XFERED mod 16; !CALCULATE THE COMP BIT
  BUF_SAV = .IO_BUF [.COMP_WRD]; !LOAD THE COMP WORD INTO BUF_SAV
  IO_BUF [.COMP_WRD] = ZERDES; !ZERO THE COMP WORD

  if .BUF_SAV [.COMP_BIT] IS_NOT_SET then return ZERO else return ONE;

!FIND COMPLIMENTED BIT AND RETURN ERROR STATUS
end;
```

3475 017660
 3476 017660 004167 164654
 3477 017664 005746
 3478 017666 016646 000012
 3479 017672 012746 000020
 3480 017676 004767 164544
 3481 017702 010001

.SBTTL FIND.COMP.BIT ROUTINE DECLARATION SECTION

```

FIND.COMP.BIT:
  JSR R1,$SAVE2 :
  TST -(SP) :
  MOV 12(SP),-(SP) : BITS.XFERED,*
  MOV #20,-(SP) :
  JSR PC,BLS$DIV :
  MOV R0,R1 : *,COMP.WRD
```

2552
 2580

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 BLISS-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (10)

3526 :ML4AD
 3527 :
 3528 :
 3529 :
 3530 :
 3531 :
 3532 :
 3533 :
 3534 :
 3535 :
 3536 :
 3537 :
 3538 :
 3539 :
 3540 :
 3541 :
 3542 :
 3543 :
 3544 :
 3545 :
 3546 :
 3547 :
 3548 :
 3549 :
 3550 :
 3551 :
 3552 :
 3553 :
 3554 :
 3555 :
 3556 :
 3557 :
 3558 :
 3559 :
 3560 :
 3561 :
 3562 :
 3563 :
 3564 :
 3565 :
 3566 :
 3567 :
 3568 :
 3569 :
 3570 :
 3571 :
 3572 :
 3573 :
 3574 :
 3575 :
 3576 :
 3577 :
 3578 :
 3579 :
 3580 :

ROUTINE DECLARATION SECTION

2589 routine FORCE_REM (PLOG, CHAN) : novalue =
 2590 begin

2591
 2592 !++
 2593 FUNCTIONAL DESCRIPTION:
 2594 VIA ECC DM AND THE PRECALCULATED MODULO 2
 2595 REMAINDERS STORED IN REM_TBL[], THE
 2596 CRC_A, CRC_B AND PAR_CRC_WRD
 2597 ARE FORCED WITH ERRONEOUS CRC DATA TO
 2598 FORCE PERDICTABLE ECC ERRORS

2600 FORMAL PARAMETERS:
 2601 PLOG
 2602 POINTS TO ONE OF THE SIX WORDS OF
 2603 A CRC GROUP
 2604
 2605 CHAN
 2606 POINTS TO ONE OF 36 CHANNELS IN
 2607 A CRC GROUP

2608
 2609 IMPLICIT INPUTS:
 2610 --

2611 local
 2612 TMP_E1, !TEMP STORAGE FOR E1 DATA WORD
 2613 ALOG, !INDEX INTO REM_TBL
 2614 BLOG; !INDEX INTO REM_TBL
 2615
 2616
 2617 ALOG = .PLOG + .CHAN; !CALCULATE A_LOG
 2618 BLOG = .PLOG + .CHAN*2; !CALCULATE B_LOG
 2619
 2620 while .ALOG geq 63 do !REDUCE A_LOG UNTIL < 64
 2621 ALOG = .ALOG - 63;
 2622
 2623 while .BLOG geq 63 do !REDUCE B_LOG UNTIL < 64
 2624 BLOG = .BLOG - 63;
 2625
 2626 ECC_DM = ONE; !SET ECC DIAG MODE

2627 !+
 2628 THE FOLLOWING ASSIGNMENT:
 2629
 2630 MLE1 = .TMP_E1
 2631
 2632 IS EQUIVALENT TO THE FOLLOWING
 2633 TWO ASSINGMENTS:

2634
 2635 PAR_CRC_WRD = .REM_TBL [.PLOG]; !LOAD PAR_CRC_WRD WITH REM_TBL
 2636 CRC_A = .REM_TBL [.ALOG]; !LOAD CRC_A WITH REM_TBL
 2637

2638 THIS IS NECESSARY DUE TO THE FACT THAT
 2639 THE E1 REGISTER IS WORD ORIENTATED AND
 2640 THE BLISS COMPILER GENERATES BYTE INST

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (10)

```

3582 :ML4AD
3583 :
3584 :
3585 : 2641 ! TO ACCESS THE REGISTER.
3586 : 2642 !-
3587 : 2643
3588 : 2644
3589 : 2645
3590 : 2646
3591 : 2647
3592 : 2648
3593 : 2649
3594 : 2650
3598 :
3599 :
    
```

ROUTINE DECLARATION SECTION

```

TMP_E1 = ZEROES;
TMP_E1<0, 6> = .REM_TBL [.ALOG];
TMP_E1<8, 6> = .REM_TBL [.PLOG];
MLET = .TMP_E1;

CRC_B = .REM_TBL [.BLOG];
end;
    
```

```

!CLEAR OUT THE TEMP WORD
!LOAD CRC_A WITH REM_TBL
!LOAD PAR_CRC WRD WITH REM_TBL
!LOAD THE E1 REGISTER WITH DATA
!LOAD CRC_B WITH REM_TBL
    
```

.SBTTL FORCE.REM ROUTINE DECLARATION SECTION

```

3603 020032 004167 164534 FORCE.REM: JSR R1,$SAVE4
3604 020032 016601 000016 MOV 16(SP),R1
3605 020036 016601 000016 MOV R1,R0
3606 020042 010100 ADD 14(SP),R0
3607 020044 066600 000014 MOV 14(SP),R2
3608 020050 016602 000014 ASL R2
3609 020054 006302 ADD R1,R2
3610 020056 060102 MOV R2,R3
3611 020060 010203 1$: CMP R0,#77
3612 020062 020027 000077 BLT 2$
3613 020066 002403 SUB #77,R0
3614 020070 162700 000077 BR 1$
3615 020074 000772 2$: CMP R3,#77
3616 020076 020327 000077 BLT 3$
3617 020102 002403 SUB #77,R3
3618 020104 162703 000077 BR 2$
3619 020110 000772 3$: BISB #1,@ML.REG+120
3620 020112 152777 000001 175404 CLR R2
3621 020120 005002 MOVB REM.TBL(R0),R4
3622 020122 116004 015664 BIC #177700,R4
3623 020126 042704 177700 BICB #77,R2
3624 020132 142702 000077 BIS R4,R2
3625 020136 050402 MOVB REM.TBL(R1),R4
3626 020140 116104 015664 SWAB R4
3627 020144 000304 BIC #140377,R4
3628 020146 042704 140377 BIC #37400,R2
3629 020152 042702 037400 BIS R4,R2
3630 020156 050402 MOV R2,@ML.REG+150
3631 020160 010277 175370 MOVB REM.TBL(R3),R4
3632 020164 116304 015664 BIC #177700,R4
3633 020170 042704 177700 BICB #77,@ML.REG+160
3634 020174 142777 000077 175362 BISB R4,@ML.REG+160
3635 020202 150477 175356 BICB R4,@ML.REG+160
3636 020206 000207 RTS PC
    
```

2589
 2617
 2618
 2620
 2621
 2620
 2623
 2624
 2623
 2626
 2644
 2645
 2646
 2647
 2649
 2589

3638
3639
3640
3641
3642
3643
3648
3649

:ML4AD
:

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

: Routine Size: 55 words
: Maximum stack depth per invocation: 5 words

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (11)

3651 :ML4AD
3652 :
3653 :
3654 :
3655 :
3656 :
3657 :
3658 :
3659 :
3660 :
3661 :
3662 :
3663 :
3664 :
3665 :
3666 :
3667 :
3668 :
3669 :
3670 :
3671 :
3672 :
3673 :
3674 :
3675 :
3676 :
3677 :
3678 :
3679 :
3680 :
3681 :
3682 :
3683 :
3684 :
3685 :
3686 :
3687 :
3688 :
3689 :
3690 :
3691 :
3692 :
3693 :
3694 :
3695 :
3696 :
3697 :
3698 :
3699 :
3700 :
3701 :
3702 :
3703 :
3704 :
3705 :

ROUTINE DECLARATION SECTION

2651 routine FOR_ECC_ERR (SGL, UNC) : novalue =
2652 begin

2654 !++

2655 FUNCTIONAL DESCRIPTION:

2656 'FORCE ECC ERROR' WHEN CALLED WILL FORCE
2657 VIA ECC DIAG MODE, SINGLE BIT ECC ERRORS
2658 OR UNCORRECTABLE ECC ERRORS INTO THE
2659 ECC DIAGNOSTIC REGISTERS

2661 FORMAL PARAMETERS:

2662 SGL
2663 WHEN SET TO A ONE THIS
2664 ROUTINE WILL FORCE SINGLE BIT ERRORS

2665 UNC
2666 WHEN SET TO A ONE THIS ROUTINE WILL
2667 FORCE UNCORRECTABLE ECC ERRORS

2670 IMPLICIT INPUTS:

2671 --

2673 local

2674 TMP_E1;

!TEMP STORAGE FOR E1 DATA WORD

2676 ECC_DM = ONE;

!SET ECC DIAG MODE

2678 if .SGL

!IF SGL IS TRUE

2679 then

2680 begin

!THEN FORCE A SINGLE BIT ERROR

2681 +

2682 THE FOLLOWING ASSIGNMENT:

2684 MLE1 = .TMP_E1

2686 IS EQUIVALENT TO THE FOLLOWING
2687 TWO ASSINGMENTS:

2689 PAR_CRC_WRD = %b'111111';

!LOAD PAR_CRC_WRD WITH REM_TBL

2690 CRC_A = %b'000000';

!LOAD CRC_A WITH REM_TBL

2692 THIS IS NECESSARY DUE TO THE FACT THAT
2693 THE E1 REGISTER IS WORD ORIENTATED AND
2694 THE BLISS COMPILER GENERATES BYTE INST
2695 TO ACCESS THE REGISTER.

2697 TMP_E1 = ZEROES;

!CLEAR OUT THE TEMP WORD

2699 TMP_E1<0, 6> = %b'000000';

!LOAD CRC_A WITH REM_TBL

2700 TMP_E1<8, 6> = %b'111111';

!LOAD PAR_CRC_WRD WITH REM_TBL

2702 MLET = .TMP_E1;

!LOAD THE E1 REGISTER WITH DATA

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (11)

3707 :ML4AD
 3708 :
 3709 :
 3710 :
 3711 :
 3712 :
 3713 :
 3714 :
 3715 :
 3716 :
 3717 :
 3718 :
 3719 :
 3720 :
 3721 :
 3722 :
 3723 :
 3724 :
 3725 :
 3726 :
 3727 :
 3728 :
 3729 :
 3730 :
 3731 :
 3732 :
 3733 :
 3734 :
 3735 :
 3736 :
 3737 :
 3738 :
 3739 :
 3743 :
 3744 :
 3748 020210
 3749 020210
 3750 020216
 3751 020220
 3752 020226
 3753 020230
 3754 020234
 3755 020236
 3756 020242
 3757 020246
 3758 020252
 3759 020260
 3760 :
 3761 :

ROUTINE DECLARATION SECTION

```

    2703     CRC_B = %b'000000';
    2704     end
    2705     else
    2706     begin
    2707     +
    2708     THE FOLLOWING ASSIGNMENT:
    2709
    2710         MLE1 = .TMP_E1
    2711
    2712     IS EQUIVALENT TO THE FOLLOWING
    2713     TWO ASSINGMENTS:
    2714
    2715         PAR_CRC_WRD = %b'111111';
    2716         CRC_A = %b'111111';
    2717
    2718     THIS IS NECESSARY DUE TO THE FACT THAT
    2719     THE E1 REGISTER IS WORD ORIENTATED AND
    2720     THE BLISS COMPILER GENERATES BYTE INST
    2721     TO ACCESS THE REGISTER.
    2722
    2723
    2724         TMP_E1 = ZEROES;
    2725         TMP_E1<0, 6> = %b'111111';
    2726         TMP_E1<8, 6> = %b'111111';
    2727         MLET = .TMP_E1;
    2728
    2729         CRC_B = %b'000000';
    2730     end;
    2731
    2732     end;
    
```

!ELSE FORCE A UNCORRECTABLE ERROR

!LOAD PAR_CRC_WRD WITH REM_TBL
 !LOAD CRC_A WITH REM_TBL

!CLEAR OUT THE TEMP WORD
 !LOAD CRC_A WITH REM_TBL
 !LOAD PAR_CRC_WRD WITH REM_TBL
 !LOAD THE E1 REGISTER WITH DATA

SBTTL FOR.ECC.ERR ROUTINE DECLARATION SECTION

```

FOR.ECC.ERR:
    BISB #1,2ML.REG+120
    CLR R0
    BIT #1,4(SP)
    BEQ 1$
    BICB #77,R0
    BR 2$
    1$: BISB #77,R0
    2$: BIS #37400,R0
    MOV R0,2ML.REG+150
    BICB #77,2ML.REG+160
    RTS PC
    
```

2676
 2698
 2678
 2699
 2700
 2725
 2726
 2727
 2729
 2651

: Routine Size: 21 words

3763
3764
3765
3766
3771
3772

;ML4AD
;

ROUTINE DECLARATION SECTION

; Maximum stack depth per invocation: 0 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (12)

3774 ;ML4AD

3775 :
 3776 :
 3777 : 2733
 3778 : 2734
 3779 : 2735
 3780 : 2736
 3781 : 2737
 3782 : 2738
 3783 : 2739
 3784 : 2740
 3785 : 2741
 3786 : 2742
 3787 : 2743
 3788 : 2744
 3789 : 2745
 3790 : 2746
 3791 : 2747
 3792 : 2748
 3793 : 2749
 3794 : 2750
 3795 : 2751
 3796 : 2752
 3797 : 2753
 3798 : 2754

ROUTINE DECLARATION SECTION
 routine FIRST_BLK_XFER : novalue =

```

  ++
  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
ECC DIS = ONE;
MLDA = ZEROES;
MLBA = IO BUF;
MLWC = not 255;
end;
  
```

```

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

3802
 3803
 3807 020262
 3808 020262
 3809 020270
 3810 020274
 3811 020302
 3812 020310
 3813
 3814
 3815
 3820
 3821

.SBTTL FIRST.BLK.XFER ROUTINE DECLARATION SECTION

```

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

2750
 2751
 2752
 2753
 2733

```

; Routine Size: 12 words
; Maximum stack depth per invocation: 0 words
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (13)

```

3823 :ML4AD
3824 :
3825 :
3826 : 2755 routine GD_BLK_XFER : novalue =
3827 : 2756
3828 : 2757
3829 : 2758
3830 : 2759
3831 : 2760
3832 : 2761
3833 : 2762
3834 : 2763
3835 : 2764
3836 : 2765
3837 : 2766
3838 : 2767
3839 : 2768
3840 : 2769
3841 : 2770
3842 : 2771
3843 : 2772
3844 : 2773
3845 : 2774
3846 : 2775
3847 : 2776
3848 : 2777
3849 : 2778
    
```

```

ROUTINE DECLARATION SECTION

++
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
    
```

```

3853
3854
3858 020312
3859 020312 152777 000002 175204
3860 020320 016777 173462 175106
3861 020326 012777 014022 175070
3862 020334 012777 177400 175052
3863 020342 000207
3864
3865
3866
3871
3872

.SBTTL GD.BLK.XFER ROUTINE DECLARATION SECTION
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

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

2774
 2775
 2776
 2777
 2755

29-Mar-1982 16:23:04 TOPS-20 BLISS-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (14)

3874 :ML4AD
 3875 :
 3876 :
 3877 :
 3878 :
 3879 :
 3880 :
 3881 :
 3882 :
 3883 :
 3884 :
 3885 :
 3886 :
 3887 :
 3888 :
 3889 :
 3890 :
 3891 :
 3892 :
 3893 :
 3894 :
 3895 :
 3896 :
 3897 :
 3898 :
 3899 :
 3903 :
 3904 :
 3908 020344
 3909 020344
 3910 020352
 3911 020360
 3912 020366
 3913 020374
 3914 :
 3915 :
 3916 :
 3921 :
 3922 :

ROUTINE DECLARATION SECTION

```

2779 routine LAST_BLK_XFER : novalue =
2780
2781 !++
2782 ! FUNCTIONAL DESCRIPTION:
2783 ! A REPEATEDLY CALLED SEQUENCE OF
2784 ! ASSIGNMENT EXPRESSIONS TO LOAD
2785 ! THE DSA, BUS ADRS AND WORD
2786 ! COUNT REGISTERS WITH APPROPRIATE
2787 ! INFORMATION BEFORE A MBUS
2788 ! TRANSFER CAN COMMENCE
2789
2790 ! LOADS A MASS BUS BLOCK
2791 ! TRANSFER AT THE LAST BLOCK
2792 ! ADDRESS
2793
2794 !--
2795
2796 begin
2797 ECC DIS = ONE;
2798 MLDA = .LST_BLK;
2799 MLBA = IO_BUF;
2800 MLWC = not 255;
2801 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 256
    
```

.SBTTL LAST.BLK.XFER ROUTINE DECLARATION SECTION

```

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
    
```

2797
 2798
 2799
 2800
 2779

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (15)

3924 :ML4AD
3925 :
3926 :
3927 :
3928 :
3929 :
3930 :
3931 :
3932 :
3933 :
3934 :
3935 :
3936 :
3937 :
3938 :
3939 :
3940 :
3941 :
3942 :
3943 :
3944 :
3945 :
3946 :
3947 :
3948 :
3949 :
3950 :
3951 :
3952 :
3953 :
3954 :
3955 :
3956 :
3957 :
3958 :
3959 :
3960 :
3961 :
3962 :
3963 :
3964 :
3965 :
3966 :
3967 :
3968 :
3969 :
3970 :
3971 :
3972 :
3973 :
3974 :
3975 :
3976 :
3977 :
3978 :

ROUTINE DECLARATION SECTION

2802 routine TST_LNG_WRD (NIB_NUM, NIB_PAT, ERR_FLG) : novalue =
2803 begin

2804
2805 !++

FUNCTIONAL DESCRIPTION:

2806 COMPARES THE CURRENT NIBBLE
2807 POSITION IN 'NIB_SAVE' WITH THE
2808 CURRENT TEST PATTERN. IF THE
2809 TWO VALUES ARE NOT EQUAL AN
2810 ERROR FLG IS SET WHICH THE
2811 CALLER CAN INTERIGATE

2812 FORMAL PARAMETERS:

2813 NIB_NUM
2814 CASE SELECT EXPRESSION TO SELECT THE
2815 CURRENT NIBBLE TO BE EXAMINED

2816 NIB_PAT
2817 CURRENT NIBBLE PATTERN TO BE
2818 COMPARED

2819 ERR_FLG
2820 CONTAINS THE ADDRESS (PASSED BY REF)
2821 OF THE CALLERS ERROR FLG
2822 TO ENABLE THE CALLER TO EXAMINE
2823 THE ERROR STATUS OF THE ROUTINE CALL

2824 IMPLICIT INPUTS:

2825 NIB_SAVE
2826 BLOCK OF 3 WORDS TO STORE
2827 THE DATA FOUND IN MLD1, MLD2
2828 AND MLE2 AFTER A DIAGNOSTIC MODE
2829 READ

2830 IMPLICIT OUTPUTS: NONE

2831 .ERR_FLG = ZERO;

!CLEAR THE ERROR FLAG BACK IN THE CALLING ROUTINE

2832 case .(NIB_NUM) from 0 to 9 of
2833 set

!SELECT THE NIBBLE TO BE TESTED

2834 [0] :

2835 if .NIB_SAVE [NIB_0] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

!TEST NIBBLE 0 AND SET ERR FLG IF NEQ

2836 [1] :

2837 if .NIB_SAVE [NIB_1] neq .(NIB_PAT)<0, 4> then .ERR_FLG = ONE;

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (15)

3980 :ML4AD
3981 :
3982 :
3983 :
3984 :
3985 :
3986 :
3987 :
3988 :
3989 :
3990 :
3991 :
3992 :
3993 :
3994 :
3995 :
3996 :
3997 :
3998 :
3999 :
4000 :
4001 :
4002 :
4003 :
4004 :
4005 :
4006 :
4007 :
4008 :
4009 :
4010 :
4011 :
4012 :
4013 :
4014 :
4015 :
4016 :
4017 :
4018 :
4019 :
4020 :
4021 :
4022 :
4023 :
4024 :
4025 :
4026 :
4027 :
4028 :
4029 :
4030 :
4031 :
4032 :
4033 :
4034 :

ROUTINE DECLARATION SECTION

2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905

[2] :

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

!TEST NIBBLE 1 AND SET ERR FLG IF NEQ

!TEST NIBBLE 2 AND SET ERR FLG IF NEQ

[3] :

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

!TEST NIBBLE 3 AND SET ERR FLG IF NEQ

[4] :

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

!TEST NIBBLE 4 AND SET ERR FLG IF NEQ

[5] :

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

!TEST NIBBLE 5 AND SET ERR FLG IF NEQ

[6] :

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

!TEST NIBBLE 6 AND SET ERR FLG IF NEQ

[7] :

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

!TEST NIBBLE 7 AND SET ERR FLG IF NEQ

[8] :

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

!TEST NIBBLE 8 AND SET ERR FLG IF NEQ

[9] :

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

.ERR_FLG = ONE

!TEST NIBBLE 9 AND SET ERR FLG IF NEQ

tes;

end;

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (16)

4179 :ML4AD
4180 :
4181 :
4182 :
4183 :
4184 :
4185 :
4186 :
4187 :
4188 :
4189 :
4190 :
4191 :
4192 :
4193 :
4194 :
4195 :
4196 :
4197 :
4198 :
4199 :
4200 :
4201 :
4202 :
4203 :
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 :

ROUTINE DECLARATION SECTION

routine XOR_LNG_WRD (NIB_NUM, NIB_PAT, RESULT) : novalue =
begin

!++

FUNCTIONAL DESCRIPTION:

EXCLUSIVE ORS THE CURRENT
NIBBLE POSITION IN 'NIB_SAVE'
WITH THE CURRENT TEST PATTERN
AND ASSIGNS THE RESULTS TO THE
FORMAL PARAMETER 'RESULT'.

FORMAL PARAMETERS:

NIB_NUM
CASE SELECT EXPRESSION TO
SELECT THE CURRENT NIBBLE TO BE
EXAMINED

NIB_PAT
CURRENT NIBBLE PATTERN TO BE
XOR'ED

RESULT
CONTAINS THE ADDRESS (PASSED BY REF)
OF AN OWN STORAGE LOCATION TO
ENABLE THE CALLER TO EXAMINE THE XOR RESULTS.

IMPLICIT INPUTS:

NIB_SAVE
BLOCK OF 3 WORDS TO STORE
THE DATA FOUND IN MLD1
MLD2 AND MLE2 AFTER A
DIAGNOSTIC MODE READ.

IMPLICIT OUTPUTS: NONE

case .(NIB_NUM) from 0 to 9 of
set

!SELECT THE NIBBLE TO BE XOR'ED

[0] :
.RESULT = .NIB_SAVE [NIB_0] xor .NIB_PAT;
!XOR NIBBLE 0 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[1] :
.RESULT = .NIB_SAVE [NIB_1] xor .NIB_PAT;
!XOR NIBBLE 1 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[2] :
.RESULT = .NIB_SAVE [NIB_2] xor .NIB_PAT;
!XOR NIBBLE 2 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS

[3] :

4235 :ML4AD
4236 :
4237 :
4238 :
4239 :
4240 :
4241 :
4242 :
4243 :
4244 :
4245 :
4246 :
4247 :
4248 :
4249 :
4250 :
4251 :
4252 :
4253 :
4254 :
4255 :
4256 :
4257 :
4258 :
4259 :
4260 :
4261 :
4262 :
4263 :
4264 :
4265 :
4266 :
4270 :
4271 :

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (16)

```

2958 .RESULT = .NIB_SAVE [NIB_3] xor .NIB PAT;
2959 !XOR NIBBLE 3 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[4] :
2961 .RESULT = .NIB_SAVE [NIB_4] xor .NIB PAT;
2962 !XOR NIBBLE 4 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[5] :
2964 .RESULT = .NIB_SAVE [NIB_5] xor .NIB PAT;
2965 !XOR NIBBLE 5 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[6] :
2967 .RESULT = .NIB_SAVE [NIB_6] xor .NIB PAT;
2968 !XOR NIBBLE 6 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[7] :
2970 .RESULT = .NIB_SAVE [NIB_7] xor .NIB PAT;
2971 !XOR NIBBLE 7 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[8] :
2973 .RESULT = .NIB_SAVE [NIB_8] xor .NIB PAT;
2974 !XOR NIBBLE 8 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
[9] :
2976 .RESULT = .NIB_SAVE [NIB_9] xor .NIB PAT;
2977 !XOR NIBBLE 9 AND STORE RESULTS IN THE ADRS CONTAINED IN .RESULTS
tes:
2978
2979
2980
2981
2982
2983
2984
2985
2986
end:

```

4275 021072
4276 021072 004167 163474
4277 021076 016601 000016
4278 021102 016602 000014
4279 021106 016600 000020
4280 021112 006300
4281 021114 066007 021120
4282 021120 000024
4283 021122 000036
4284 021124 000062
4285 021126 000070
4286 021130 000076
4287 021132 000120
4288 021134 000136
4289 021136 000144

.SBTTL XOR.LNG.WRD ROUTINE DECLARATION SECTION

```

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

```

2906
2946
2942

4347
4348
4349
4350 021346 040112
4351 021350 050312
4352 021352 000207
4353
4354
4355
4360
4361

:ML4AD
:
ROUTINE DECLARATION SECTION
BIC R1,(R2)
BIS R3,(R2)
RTS PC
:
: Routine Size: 89 words
: Maximum stack depth per invocation: 5 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

2906

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (17)

4363 :ML4AD
4364 :
4365 :
4366 :
4367 :
4368 :
4369 :
4370 :
4371 :
4372 :
4373 :
4374 :
4375 :
4376 :
4377 :
4378 :
4379 :
4380 :
4381 :
4382 :
4383 :
4384 :
4385 :
4386 :
4387 :
4388 :
4389 :
4390 :
4391 :
4392 :
4393 :
4394 :
4395 :
4396 :
4397 :
4398 :
4399 :
4400 :
4401 :
4402 :
4403 :
4404 :
4405 :
4406 :
4407 :
4408 :
4409 :
4410 :
4411 :
4412 :
4413 :
4414 :
4415 :
4416 :
4417 :

ROUTINE DECLARATION SECTION

routine LD_LNG_WRD (NIB_NUM, NIB_PAT) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:
LOADS 'NIB_SAVE' WITH UNIQUE
NIBBLE PATTERNS PRIOR TO WRITING
TO MLD1, MLD2 AND MLE2
DATA DIAGNOSTIC REGISTERS.

FORMAL PARAMETERS:
NIB_NUM
CASE SELECT EXPRESSION TO SELECT
THE CURRENT NIBBLE TO BE LOADED

NIB_PAT
CURRENT NIBBLE PATTERN TO BE
LOADED

IMPLICIT INPUTS:
NIB_SAVE
BLOCK OF 3 WORDS TO STORE
THE DATA TO BE WRITTEN
INTO MLD1 MLD2 MLE2

IMPLICIT OUTPUTS:
NIB_SAVE IS LOADED WITH
THE CURRENT NIBBLE PATTERN

!--
case .(NIB_NUM) from 0 to 9 of
set
[0] : !SELECT THE NIBBLE LOCATION IN NIB_SAVE TO BE LOADED
NIB_SAVE [NIB_0] = .NIB_PAT; !LOAD NIBBLE 0 WITH NIB_PAT
[1] : !LOAD NIBBLE 1 WITH NIB_PAT
NIB_SAVE [NIB_1] = .NIB_PAT;
[2] : !LOAD NIBBLE 2 WITH NIB_PAT
NIB_SAVE [NIB_2] = .NIB_PAT;
[3] : !LOAD NIBBLE 3 WITH NIB_PAT
NIB_SAVE [NIB_3] = .NIB_PAT;
[4] : !LOAD NIBBLE 4 WITH NIB_PAT
NIB_SAVE [NIB_4] = .NIB_PAT;
[5] : !LOAD NIBBLE 5 WITH NIB_PAT
NIB_SAVE [NIB_5] = .NIB_PAT;
[6] :

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (17)

```

4419 ;ML4AD
4420 :
4421 :
4422 : 3039 NIB_SAVE [NIB_6] = .NIB_PAT; !LOAD NIBBLE 6 WITH NIB_PAT
4423 : 3040
4424 : 3041 [7] :
4425 : 3042 NIB_SAVE [NIB_7] = .NIB_PAT; !LOAD NIBBLE 7 WITH NIB_PAT
4426 : 3043
4427 : 3044 [8] :
4428 : 3045 NIB_SAVE [NIB_8] = .NIB_PAT; !LOAD NIBBLE 8 WITH NIB_PAT
4429 : 3046
4430 : 3047 [9] :
4431 : 3048 NIB_SAVE [NIB_9] = .NIB_PAT !LOAD NIBBLE 9 WITH NIB_PAT
4432 : 3049
4433 : 3050 tes:
4434 : 3051 end:
  
```

```

4439
4443 021354 .SBTTL LD.LNG.WRD ROUTINE DECLARATION SECTION
4444 021354 010146 LD.LNG.WRD:
4445 021356 016600 000004 MOV R1,-(SP) ;
4446 021362 016601 000006 MOV 4(SP),R0 ; NIB.PAT,*
4447 021366 006301 MOV 6(SP),R1 ; NIB.NUM,*
4448 021370 066107 021374 ASL R1
4449 021374 000024 1$: ADD 1$(R1),PC
4450 021376 000040 .WORD 2$-1$
4451 021400 000064 .WORD 3$-1$
4452 021402 000102 .WORD 4$-1$
4453 021404 000134 .WORD 5$-1$
4454 021406 000150 .WORD 6$-1$
4455 021410 000174 .WORD 7$-1$
4456 021412 000212 .WORD 8$-1$
4457 021414 000244 .WORD 9$-1$
4458 021416 000262 .WORD 10$-1$
4459 021420 042700 177760 2$: BIC #177760,R0 ;
4460 021424 142767 000017 171742 BICB #17,NIB.SAVE ;
4461 021432 000433 BR 6$ ;
4462 021434 006300 3$: ASL R0 ;
4463 021436 006300 ASL R0 ;
4464 021440 006300 ASL R0 ;
4465 021442 006300 ASL R0 ;
4466 021444 042700 177417 BIC #177417,R0 ;
4467 021450 142767 000360 171716 BICB #360,NIB.SAVE ;
4468 021456 000421 BR 6$ ;
4469 021460 000300 4$: SWAB R0 ;
4470 021462 042700 170377 BIC #170377,R0 ;
4471 021466 042767 007400 171700 BICB #7400,NIB.SAVE ;
4472 021474 000412 BR 6$ ;
4473 021476 000300 5$: SWAB R0 ;
  
```

2987
 3021
 3017
 3021
 3024
 3027
 3030

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (18)

4531 :ML4AD
 4532 :
 4533 :
 4534 :
 4535 :
 4536 :
 4537 :
 4538 :
 4539 :
 4540 :
 4541 :
 4542 :
 4543 :
 4544 :
 4545 :
 4546 :
 4547 :
 4548 :
 4549 :
 4550 :
 4551 :
 4552 :
 4553 :
 4554 :
 4558 :
 4559 :
 4563 021712 004167 162622
 4564 021716 016600 000010
 4565 021722 006300
 4566 021724 006300
 4567 021726 006300
 4568 021730 010001
 4569 021732 016100 015406
 4570 021736 056600 000012
 4571 021742 046100 015410
 4572 021746 016102 015412
 4573 021752 050002
 4574 021754 010277 173424
 4575 021760 000207
 4576 :
 4577 :
 4578 :
 4583 :
 4584 :

ROUTINE DECLARATION SECTION

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

WRT_CS1: .SBTTL WRT_CS1 ROUTINE DECLARATION SECTION

```

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

3052
 3071

3052

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (19)

4586 :ML4AD
4587 :
4588 :
4589 :
4590 :
4591 :
4592 :
4593 :
4594 :
4595 :
4596 :
4597 :
4598 :
4599 :
4600 :
4601 :
4602 :
4603 :
4604 :
4605 :
4606 :
4607 :
4608 :
4609 :
4610 :
4611 :
4612 :
4613 :
4614 :
4615 :
4616 :
4617 :
4618 :
4619 :
4620 :
4621 :
4622 :
4623 :
4624 :
4625 :
4626 :
4627 :
4628 :
4629 :
4630 :
4631 :
4632 :
4633 :
4634 :
4635 :
4636 :
4637 :
4638 :
4639 :
4640 :

ROUTINE DECLARATION SECTION

routine RD_CS1 (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
CONTROL & STATUS 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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA.<NEALE>ML4AD.BLI.4 (19)

4642 :ML4AD
4643 :
4644 :
4645 :
4646 :
4647 :
4648 :
4649 :
4650 :
4651 :
4652 :

ROUTINE DECLARATION SECTION

```
.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

4656
4657
4661 021762 004167 162552
4662 021766 005076 000010
4663 021772 016600 000012
4664 021776 006300
4665 022000 006300
4666 022002 006300
4667 022004 010001
4668 022006 016100 015406
4669 022012 056600 000014
4670 022016 046100 015410
4671 022022 016102 015412
4672 022026 050002
4673 022030 010267 173314
4674 022034 017702 173344
4675 022040 056102 015412
4676 022044 010267 173302
4677 022050 026767 173274 173274
4678 022056 001403
4679 022060 012776 000001 000010
4680 022066 000207
4681
4682
4683
4688
4689

```
.SBTTL RD.CS1 ROUTINE DECLARATION SECTION
RD.CS1: JSR R1,$SAVE2
CLR @10(SP)
MOV 12(SP),R0
ASL R0
ASL R0
ASL R0
MOV R0,R1
MOV ML.REG+2(R1),R0
BIS 14(SP),R0
BIC ML.REG+4(R1),R0
MOV ML.REG+6(R1),R2
BIS R0,R2
MOV R2,WT_DATA
MOV @ML.REG,R2
BIS ML.REG+6(R1),R2
MOV R2,RD_DATA
CMP WT_DATA,RD_DATA
BEQ 1$
MOV #1,@10(SP)
1$: RTS PC
```

3073
3125
3126

3127

3129

3073

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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (20)

4691 :ML4AD
 4692 :
 4693 :
 4694 :
 4695 :
 4696 :
 4697 :
 4698 :
 4699 :
 4700 :
 4701 :
 4702 :
 4703 :
 4704 :
 4705 :
 4706 :
 4707 :
 4708 :
 4709 :
 4710 :
 4711 :
 4712 :
 4713 :
 4717 :
 4718 :
 4722 022070 004167 162444
 4723 022074 016600 000010
 4724 022100 006300
 4725 022102 006300
 4726 022104 006300
 4727 022106 010001
 4728 022110 016100 015406
 4729 022114 056600 000012
 4730 022120 046100 015410
 4731 022124 016102 015412
 4732 022130 050002
 4733 022132 010277 173326
 4734 022136 000207
 4735 :
 4736 :
 4737 :
 4742 :
 4743 :

ROUTINE DECLARATION SECTION
 routine WRT_ER (TST_PAT, index) : novalue =
 begin

!++
 FUNCTIONAL DESCRIPTION:
 LOADS THE ERROR REGISTER 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.
 !--

MLER = WRT_MASK; !LOAD GENERATE WRT_MASK PATTERN INTO MLER
 end;

.SBTTL WRT_ER ROUTINE DECLARATION SECTION
 WRT_ER: JSR R1,\$SAVE2
 MOV 10(SP),R0 ; INDEX,*
 ASL R0
 ASL R0
 ASL R0
 MOV R0,R1
 MOV ML.REG+2(R1),R0
 BIS 12(SP),R0 ; TST.PAT,*
 BIC ML.REG+4(R1),R0
 MOV ML.REG+6(R1),R2
 BIS R0,R2
 MOV R2,ML.REG+60
 RTS PC ;

3133
 3151
 3133

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (21)

4745 :ML4AD
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 :
4790 :
4791 :
4792 :
4793 :
4794 :
4795 :
4796 :
4797 :
4798 :
4799 :

3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178
3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204

ROUTINE DECLARATION SECTION
routine RD_ER (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

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

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

FORMAL PARAMETERS:

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

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

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

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

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

IMPLICIT OUTPUTS:
GLOBAL LOCATION WR_DATA
AND RD_DATA LOADED WITH GOOD
AND BAD REGISTER DATA

--

4801 :ML4AD
 4802 :
 4803 :
 4804 : 3205
 4805 : 3206
 4806 : 3207
 4807 : 3208
 4808 : 3209
 4809 : 3210
 4810 : 3211
 4811 : 3212
 4812 : 3213
 4816 :
 4817 :
 4821 022140 004167 162374
 4822 022144 005076 000010
 4823 022150 016600 000012
 4824 022154 006300
 4825 022156 006300
 4826 022160 006300
 4827 022162 010001
 4828 022164 016100 015406
 4829 022170 056600 000014
 4830 022174 046100 015410
 4831 022200 016102 015412
 4832 022204 050002
 4833 022206 010267 173136
 4834 022212 017702 173246
 4835 022216 056102 015412
 4836 022222 010267 173124
 4837 022226 026767 173116 173116
 4838 022234 001403
 4839 022236 012776 000001 000010
 4840 022244 000207
 4841 :
 4842 :
 4843 :
 4848 :
 4849 :

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (21)

```

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

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

end;                                     !SET ERROR FLAG IF NEQ
  
```

RD.ER: .SBTTL RD.ER ROUTINE DECLARATION SECTION

```

JSR R1,$SAVE2
CLR @10(SP)
MOV 12(SP),R0
ASL R0
ASL R0
ASL R0
MOV R0,R1
MOV ML.REG+2(R1),R0
BIS 14(SP),R0
BIC ML.REG+4(R1),R0
MOV ML.REG+6(R1),R2
BIS R0,R2
MOV R2,WT_DATA
MOV @ML.REG+60,R2
BIS ML.REG+6(R1),R2
MOV R2,RD_DATA
CMP WT_DATA,RD_DATA
BEQ 1$
MOV #1,@10(SP)
RTS PC
  
```

3153
 3206
 3207
 3208
 3210
 3153

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (22)

4851 :ML4AD
4852 :
4853 :
4854 :
4855 :
4856 :
4857 :
4858 :
4859 :
4860 :
4861 :
4862 :
4863 :
4864 :
4865 :
4866 :
4867 :
4868 :
4869 :
4870 :
4871 :
4872 :
4873 :
4877 :
4878 :

3214
3215
3216
3217
3218
3219
3220
3221
3222
3223
3224
3225
3226
3227
3228
3229
3230
3231
3232
3233

ROUTINE DECLARATION SECTION

routine WRT_DA (TST_PAT, index) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:

LOADS THE DESIRED SECTOR 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.

MLDA = WRT_MASK;
end;

!LOAD MLDA WITH GENERATED WRT_MASK PATTERN

4882 022246 004167 162266
4883 022252 016600 000010
4884 022256 006300
4885 022260 006300
4886 022262 006300
4887 022264 010001
4888 022266 016100 015406
4889 022272 056600 000012
4890 022276 046100 015410
4891 022302 016102 015412
4892 022306 050002
4893 022310 010277 173120
4894 022314 000207
4895 :
4896 :
4897 :
4902 :
4903 :

```

.SBTTL WRT_DA ROUTINE DECLARATION SECTION
WRT_DA: JSR   R1,$SAVE2
        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+30
        RTS
    
```

3214
3232

3214

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (23)

4905 :ML4AD
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 :
4950 :
4951 :
4952 :
4953 :
4954 :
4955 :
4956 :
4957 :
4958 :
4959 :

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
3270
3271
3272
3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285

ROUTINE DECLARATION SECTION

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

++
FUNCTIONAL DESCRIPTION:

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

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

FORMAL PARAMETERS:

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

INDEX
USED BY THE MACRO WRT_MASK TO
SELECT THE CURRENT REGISTER 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

--

4961 :ML4AD
4962 :
4963 :
4964 :
4965 :
4966 :
4967 :
4968 :
4969 :
4970 :
4971 :
4975 :
4976 :
4980 022316 004167 162216
4981 022322 005076 000010
4982 022326 016600 000012
4983 022332 006300
4984 022334 006300
4985 022336 006300
4986 022340 010001
4987 022342 016100 015406
4988 022346 056600 000014
4989 022352 046100 015410
4990 022356 016102 015412
4991 022362 050002
4992 022364 010267 172760
4993 022370 017702 173040
4994 022374 056102 015412
4995 022400 010267 172746
4996 022404 026767 172740 172740
4997 022412 001403
4998 022414 012776 000001 000010
4999 022422 000207
5000
5001
5002
5007
5008

ROUTINE DECLARATION SECTION

```
.ERR_FLG = ZERO;
WT_DATA = WRT_MASK;
RD_DATA = .MLDA_r .IGNORE;

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

end;
```

```
29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (23)

!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
```

RD.DA: .SBTTL RD.DA ROUTINE DECLARATION SECTION

```
JSR R1,$SAVE2
CLR @10(SP)
MOV 12(SP),R0
ASL R0
ASL R0
ASL R0
MOV R0,R1
MOV ML.REG+2(R1),R0
BIS 14(SP),R0
BIC ML.REG+4(R1),R0
MOV ML.REG+6(R1),R2
BIS R0,R2
MOV R2,WT_DATA
MOV @ML.REG+30,R2
BIS ML.REG+6(R1),R2
MOV R2,RD_DATA
CMP WT_DATA,RD_DATA
BEQ 1$
MOV #1,@10(SP)
RTS PC
```

3234
3286
3287

3288
3290
3234

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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (24)

5010 :ML4AD
 5011 :
 5012 :
 5013 :
 5014 :
 5015 :
 5016 :
 5017 :
 5018 :
 5019 :
 5020 :
 5021 :
 5022 :
 5023 :
 5024 :
 5025 :
 5026 :
 5027 :
 5028 :
 5029 :
 5030 :
 5031 :
 5032 :
 5036 :
 5037 :
 5041 022424 004167 162110
 5042 022430 016600 000010
 5043 022434 006300
 5044 022436 006300
 5045 022440 006300
 5046 022442 010001
 5047 022444 016100 015406
 5048 022450 056600 000012
 5049 022454 046100 015410
 5050 022460 016102 015412
 5051 022464 050002
 5052 022466 010277 173032
 5053 022472 000207
 5054 :
 5055 :
 5056 :
 5061 :
 5062 :

ROUTINE DECLARATION SECTION

routine WRT_MR (TST_PAT, index) : novalue =
 begin

!++

FUNCTIONAL DESCRIPTION:

LOADS THE MAINTENANCE REGISTER 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 DGN'T CARE MASK INFORMATION.

!--

MLMR = WRT_MASK;
 end;

!LOAD MLMR WITH GENERATED WRT_MASK PATTERN

```

.SBTTL WRT_MR ROUTINE DECLARATION SECTION
WRT_MR: JSR R1,$SAVE2
        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+120
        RTS PC
  
```

3294
 3312

3294

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (25)

5064 :ML4AD
5065 :
5066 :
5067 :
5068 :
5069 :
5070 :
5071 :
5072 :
5073 :
5074 :
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 :

3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365

ROUTINE DECLARATION SECTION

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

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
MAINTENANCE 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

--

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (25)

5120 ;ML4AD
5121 :
5122 :
5123 : 3366
5124 : 3367
5125 : 3368
5126 : 3369
5127 : 3370
5128 : 3371
5129 : 3372
5130 : 3373
5131 : 3374

ROUTINE DECLARATION SECTION

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

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

end;
```

```
!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
```

5136
5140 022474 004167 162040
5141 022500 005076 000010
5142 022504 016600 000012
5143 022510 006300
5144 022512 006300
5145 022514 006300
5146 022516 010001
5147 022520 016100 015406
5148 022524 056600 000014
5149 022530 046100 015410
5150 022534 016102 015412
5151 022540 050002
5152 022542 010267 172602
5153 022546 017702 172752
5154 022552 056102 015412
5155 022556 010267 172570
5156 022562 026767 172562 172562
5157 022570 001403
5158 022572 012776 000001 000010
5159 022600 000207
5160
5161
5162
5167
5168

```
.SBTTL RD.MR ROUTINE DECLARATION SECTION
RD.MR: JSR R1,$SAVE2
CLR @10(SP)
MOV 12(SP),R0
ASL R0
ASL R0
ASL R0
MOV R0,R1
MOV ML.REG+2(R1),R0
BIS 14(SP),R0
BIC ML.REG+4(R1),R0
MOV ML.REG+6(R1),R2
BIS R0,R2
MOV R2,WT_DATA
MOV @ML.REG+120,R2
BIS ML.REG+6(R1),R2
MOV R2,RD_DATA
CMP WT_DATA,RD_DATA
BEQ 1$
MOV #1,@10(SP)
1$: RTS PC
```

3314
3367
3368
3369
3371
3314

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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (26)

5170 ;ML4AD
 5171 :
 5172 :
 5173 :
 5174 :
 5175 :
 5176 :
 5177 :
 5178 :
 5179 :
 5180 :
 5181 :
 5182 :
 5183 :
 5184 :
 5185 :
 5186 :
 5187 :
 5188 :
 5189 :
 5190 :
 5194 :
 5195 :

3375
 3376
 3377
 3378
 3379
 3380
 3381
 3382
 3383
 3384
 3385
 3386
 3387
 3388
 3389
 3390
 3391
 3392

ROUTINE DECLARATION SECTION

routine WRT_PA (TST_PAT, index) : novalue =
 begin

!++

FUNCTIONAL DESCRIPTION:

LOADS THE PROM ADDRESS REGISTER 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.

!--

PROM_DIS = ONE;
 MLPA = WRT_MASK;
 PROM_DIS = ZERO;
 end;

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

5199 022602 004167 161732
 5200 022606 152777 000040 172710
 5201 022614 016600 000010
 5202 022620 006300
 5203 022622 006300
 5204 022624 006300
 5205 022626 010001
 5206 022630 016100 015406
 5207 022634 056600 000012
 5208 022640 046100 015410
 5209 022644 016102 015412
 5210 022650 050002
 5211 022652 010277 172626
 5212 022656 142777 000040 172640
 5213 022664 000207
 5214
 5215
 5216
 5221
 5222

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

3375
 3389
 3390
 3391
 3375

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (27)

5224 :ML4AD
5225 :
5226 :
5227 :
5228 :
5229 :
5230 :
5231 :
5232 :
5233 :
5234 :
5235 :
5236 :
5237 :
5238 :
5239 :
5240 :
5241 :
5242 :
5243 :
5244 :
5245 :
5246 :
5247 :
5248 :
5249 :
5250 :
5251 :
5252 :
5253 :
5254 :
5255 :
5256 :
5257 :
5258 :
5259 :
5260 :
5261 :
5262 :
5263 :
5264 :
5265 :
5266 :
5267 :
5268 :
5269 :
5270 :
5271 :
5272 :
5273 :
5274 :
5275 :
5276 :
5277 :
5278 :

3393
3394
3395
3396
3397
3398
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
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444

ROUTINE DECLARATION SECTION

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

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
PROM 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 REGISTERS 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

--

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (27)

5280 ;ML4AD

ROUTINE DECLARATION SECTION

```

5281 :
5282 :
5283 :      3445      .ERR_FLG = ZERO;          !CLEAR THE ERROR FLAG
5284 :      3446      PROM_DIS = ONE;        !SET THE PROM DISABLE BIT
5285 :      3447      WT_DATA = WRT_MASK;     !SAVE THE DATA WRITTEN TO THE REGISTER
5286 :      3448      RD_DATA = .MLPA or .IGNORE; !READ AND SAVE THE REGISTER
5287 :      3449
5288 :      3450      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;      !READ THE REG FOR WRT_MASK
5289 :      3451
5290 :      3452      PROM_DIS = ZERO;        !SET ERROR IF NEQ
5291 :      3453      end;                    !CLEAR THE PROM DISABLE BIT
5295 :
5296 :

```

```

5300 022666 004167 161646      RD.PA: .SBTTL RD.PA ROUTINE DECLARATION SECTION
5301 022672 005076 000010      JSR   R1,$SAVE2
5302 022676 152777 000040 172620      CLR   @10(SP)
5303 022704 016600 000012      BISB #40,@ML.REG+120
5304 022710 006300              MOV   12(SP),R0
5305 022712 006300              ASL   R0
5306 022714 006300              ASL   R0
5307 022716 010001              ASL   R0
5308 022720 016100 015406      MOV   R0,R1
5309 022724 056600 000014      MOV   ML.REG+2(R1),R0
5310 022730 046100 015410      BIS   14(SP),R0
5311 022734 016102 015412      BIC   ML.REG+4(R1),R0
5312 022740 050002              MOV   ML.REG+6(R1),R2
5313 022742 010267 172402      BIS   R0,R2
5314 022746 017702 172532      MOV   R2,WT_DATA
5315 022752 056102 015412      MOV   @ML.REG+100,R2
5316 022756 010267 172370      BIS   ML.REG+6(R1),R2
5317 022762 026767 172362 172362      MOV   R2,RD_DATA
5318 022770 001403              CMP   WT_DATA,RD_DATA
5319 022772 012776 000001 000010      BEQ   1$
5320 023000 142777 000040 172516 1$:      MOV   #1,@10(SP)
5321 023006 000207              BICB #40,@ML.REG+120
5322 :
5323 :
5324 :
5329 :
5330 :

```

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

3393
3445
3446
3447

3448

3450

3452
3393

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (28)

5332 :ML4AD
 5333 :
 5334 :
 5335 :
 5336 :
 5337 :
 5338 :
 5339 :
 5340 :
 5341 :
 5342 :
 5343 :
 5344 :
 5345 :
 5346 :
 5347 :
 5348 :
 5349 :
 5350 :
 5351 :
 5352 :
 5356 :
 5357 :

ROUTINE DECLARATION SECTION

3454 routine WRT_E1 (TST_PAT, index) : novalue =
 3455 begin

```

3456
3457 !++
3458 ! FUNCTIONAL DESCRIPTION:
3459 ! LOADS THE ECC CRC WORD REG 1 WITH A DATA PATTERN GENERATED BY THE MACRO WRT_MASK
3460 ! FORMAL PARAMETERS:
3461 ! TST_PAT
3462 ! CURRENT DATA PATTERN TO BE LOADED IN THE REGISTER.
3463 ! INDEX
3464 ! USED BY THE MACRO WRT_MASK TO SELECT THE CURRENT REGISTERS ADDRESS,
3465 ! FORCED HI, FORCED LO AND DON'T CARE MASK INFORMATION.
3466 !--
  
```

```

3468 ECC_DM = ONE;
3469 MLET = WRT_MASK;
3470 ECC_DM = ZERO;
3471 end;
  
```

```

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

5361	023010	004167	161524		WRT.E1:	.SBTTL	WRT.E1 ROUTINE DECLARATION SECTION		
5362	023014	152777	000001	172502		JSR	R1,\$SAVE2	:	
5363	023022	016600	000010			BISB	#1,@ML.REG+120	:	3454
5364	023026	006300				MOV	10(SP),R0	:	3468
5365	023030	006300				ASL	R0	:	3469
5366	023032	006300				ASL	R0	:	
5367	023034	010001				MOV	R0,R1	:	
5368	023036	016100	015406			MOV	ML.REG+2(R1),R0	:	
5369	023042	056600	000012			BIS	12(SP),R0	:	
5370	023046	046100	015410			BIC	ML.REG+4(R1),R0	:	
5371	023052	016102	015412			MOV	ML.REG+6(R1),R2	:	
5372	023056	050002				BIS	R0,R2	:	
5373	023060	010277	172470			MOV	R2,@ML.REG+150	:	
5374	023064	142777	000001	172432		BICB	#1,@ML.REG+120	:	
5375	023072	000207				RTS	PC	:	3470
5376								:	3454

```

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

5377
 5378
 5383
 5384

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (29)

5386 :ML4AD
5387 :
5388 :
5389 :
5390 :
5391 :
5392 :
5393 :
5394 :
5395 :
5396 :
5397 :
5398 :
5399 :
5400 :
5401 :
5402 :
5403 :
5404 :
5405 :
5406 :
5407 :
5408 :
5409 :
5410 :
5411 :
5412 :
5413 :
5414 :
5415 :
5416 :
5417 :
5418 :
5419 :
5420 :
5421 :
5422 :
5423 :
5424 :
5425 :
5426 :
5427 :
5428 :
5429 :
5430 :
5431 :
5432 :
5433 :
5434 :
5435 :
5436 :
5437 :
5438 :
5439 :
5440 :

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
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513
3514
3515
3516
3517
3518
3519
3520
3521
3522
3523

ROUTINE DECLARATION SECTION

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'S ADDRESS,
FORCED HI, FORCED LO AND 'DON'T CARE'
MASK INFORMATION.

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

IMPLICIT INPUTS:

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

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

IMPLICIT OUTPUTS:

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

--
.ERR_FLG = ZERO;

!CLEAR THE ERROR FLAG

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (29)

5442 :ML4AD

ROUTINE DECLARATION SECTION

```

5443 :
5444 :
5445 :      3524      ECC_DM = ONE;
5446 :      3525      WT_DATA = WRT_MASK;
5447 :      3526      RD_DATA = .MLE1 or .IGNORE;
5448 :      3527
5449 :      3528      if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;
5450 :      3529
5451 :      3530      ECC_DM = ZERO;
5452 :      3531
5453 :      3532      end;

```

```

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

```

```

5457 :
5458 :
5462 023074 004167 161440      RD.E1: .SBTTL RD.E1 ROUTINE DECLARATION SECTION
5463 023100 005076 000010      JSR      R1,$SAVE2
5464 023104 152777 000001 172412      CLR      @10(SP)
5465 023112 016600 000012      BISB     #1,@ML.REG+120
5466 023116 006300      MOV      12(SP),R0
5467 023120 006300      ASL     R0
5468 023122 006300      ASL     R0
5469 023124 010001      ASL     R0
5470 023126 016100 015406      MOV      R0,R1
5471 023132 056600 000014      MOV      ML.REG+2(R1),R0
5472 023136 046100 015410      BIS      14(SP),R0
5473 023142 016102 015412      BIC      ML.REG+4(R1),R0
5474 023146 050002      MOV      ML.REG+6(R1),R2
5475 023150 010267 172174      BIS      R0,R2
5476 023154 017702 172374      MOV      R2,WT_DATA
5477 023160 056102 015412      MOV      @ML.REG+150,R2
5478 023164 010267 172162      BIS      ML.REG+6(R1),R2
5479 023170 026767 172154 172154      MOV      R2,RD_DATA
5480 023176 001403      CMP      WT_DATA,RD_DATA
5481 023200 012776 000001 000010      BEQ      1$
5482 023206 142777 000001 172310 1$:      MOV      #1,@10(SP)
5483 023214 000207      BICB     #1,@ML.REG+120
5484 :      RTS      PC

```

```

: ERR.FLG
: INDEX,*
: TST.PAT,*
:
: * ,ERR.FLG
:
3472
3523
3524
3525
3526
3528
3530
3472

```

```

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

```

5485
5486
5491
5492

5494 :ML4AD

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(2'2)
 PA:<NEALE>ML4AD.BLI.4 (30)

ROUTINE DECLARATION SECTION

5495 :
 5496 :
 5497 : 3533 routine WRT_E2 (TST_PAT, index) : novalue =
 5498 : 3534 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.

5522 : 3558 ECC_DM = ONE;
 5523 : 3559 MLE2_MASK = %0'177700';
 5524 : 3560 MLE2 = WRT_MASK;
 5525 : 3561 MLE2_MASK = %0'000000';
 5526 : 3562 ECC_DM = ZERO;
 5527 : 3563 end;

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

```

5531 :
5532 :
5536 023216 004167 161316      .SBTTL  WRT.E2 ROUTINE DECLARATION SECTION
5537 023222 152777 000001 172274 WRT.E2: JSR   R1,$SAVE2
5538 023230 012767 177700 172334     BISB  #1,2(R1),REG+120
5539 023236 016600 000010     MOV   #-100,ML,REG+166
5540 023242 006300           MOV   10(SP),R0
5541 023244 006300           ASL  R0
5542 023246 006300           ASL  R0
5543 023250 010001           MOV   R0,R1
5544 023252 016100 015406     MOV   ML,REG+2(R1),R0
5545 023256 056600 000012     BIS  12(SP),R0
5546 023262 046100 015410     BIC  ML,REG+4(R1),R0
5547 023266 016102 015412     MOV  ML,REG+6(R1),R2
5548 023272 050002           BIS  R0,R2
  
```

3533
 3558
 3559
 3560


```
5550                                   :ML4AD  
5551                                   :  
5552                                   : ROUTINE DECLARATION SECTION  
5553 023274 010277 172264           MOV    R2,@ML.REG+160  
5554 023300 005067 172266           CLR    ML.REG+166  
5555 023304 142777 000001 172212   BICB  #1,@ML.REG+120  
5556 023312 000207                   RTS    PC  
5557  
5558                                   : Routine Size: 31 words  
5559                                   : Maximum stack depth per invocation: 3 words  
5564  
5565
```

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

3561
3562
3533

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (31)

5567 :ML4AD
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 :
5608 :
5609 :
5610 :
5611 :
5612 :
5613 :
5614 :
5615 :
5616 :
5617 :
5618 :
5619 :
5620 :
5621 :

ROUTINE DECLARATION SECTION

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 REGISTERS 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

-

5623 :ML4AD

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (31)

```

5624 :
5625 :
5626 : 3616
5627 : 3617 .ERR_FLG = ZERO; !CLEAR ERROR FLAG
5628 : 3618 ECC_DM = ONE; !SET ECC DIAG MODE
5629 : 3619 MLE2_MASK = %0'177700'; !MASK OUT DATA DIAG BITS
5630 : 3620 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5631 : 3621 RD_DATA = .MLE2 or .IGNORE; !READ AND SAVE THE REGISTER
5632 : 3622
5633 : 3623 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
5634 : 3624
5635 : 3625
5636 : 3626 MLE2_MASK = %0'000000'; !SET ERROR FLG IF NEQ
5637 : 3627 ECC_DM = ZERO; !RESTORE MLE2 MASK
5638 : 3628 end; !CLEAR ECC DIAG MODE

```

```

5642 :
5643 :
5647 023314 004167 161220 RD.E2: .SBTTL RD.E2 ROUTINE DECLARATION SECTION
5648 023320 005076 000010 JSR R1,$SAVE2
5649 023324 152777 000001 172172 CLR @10(SP) ; ERR.FLG
5650 023332 012767 177700 172232 BISB #1,@ML.REG+120
5651 023340 016600 000012 MOV #-100,ML.REG+166
5652 023344 006300 ASL R0 ; INDEX,*
5653 023346 006300 ASL R0
5654 023350 006300 ASL R0
5655 023352 010001 MOV R0,R1
5656 023354 016100 015406 MOV ML.REG+2(R1),R0
5657 023360 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5658 023364 046100 015410 BIC ML.REG+4(R1),R0
5659 023370 016102 015412 MOV ML.REG+6(R1),R2
5660 023374 050002 BIS R0,R2
5661 023376 010267 171746 MOV R2,WT_DATA
5662 023402 017702 172156 MOV @ML.REG+160,R2
5663 023406 056102 015412 BIS ML.REG+6(R1),R2 ;
5664 023412 010267 171734 MOV R2,RD_DATA ;
5665 023416 026767 171726 171726 CMP WT_DATA,RD_DATA ;
5666 023424 001403 BEQ 1$ ;
5667 023426 012776 000001 000010 1$: MOV #1,@10(SP) ; *.ERR.FLG
5668 023434 005067 172132 CLR ML.REG+166 ;
5669 023440 142777 000001 172056 BICB #1,@ML.REG+120 ;
5670 023446 000207 RTS PC ;
5671 :
5672 :
5673 :

```

: Routine Size: 46 words
: Maximum stack depth per invocation: 3 words

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (32)

5679 :ML4AD

ROUTINE DECLARATION SECTION

5680 :
 5681 :
 5682 : 3629 routine WRT_PD (TST_PAT, index) : novalue =
 5683 : 3630 begin

!++
 FUNCTIONAL DESCRIPTION:

LOADS THE PROM DATA
 REGISTER 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

PROM_RW = ONE; !SET PROM READ WRITE
 DAT_DM = ONE; !SET DATA DIAG MODE
 MLPD = WRT_MASK; !LOAD MLPD WITH GENERATED WRT_MASK PATTERN
 DAT_CLK = ONE; !DO A DATA CLK
 PROM_RW = ZERO; !CLEAR PROM READ WRITE
 DAT_DM = ZERO; !CLEAR DATA DIAG MODE
 end;

5718									
5719						.SBTTL	WRT_PD ROUTINE DECLARATION SECTION		
5723	023450	004167	161064			WRT_PD:	JSR R1,\$SAVE2		3629
5724	023454	152777	000100	172042			BISB #100,@ML.REG+120		3655
5725	023462	152777	000010	172034			BISB #10,@ML.REG+120		3656
5726	023470	016600	000010				MOV 10(SP),R0		3657
5727	023474	006300					ASL R0	: INDEX,*	
5728	023476	006300					ASL R0		
5729	023500	006300					ASL R0		
5730	023502	010001					MOV R0,R1		
5731	023504	016100	015406				MOV ML.REG+2(R1),R0		
5732	023510	056600	000012				BIS 12(SP),R0		
5733	023514	046100	015410				BIC ML.REG+4(R1),R0	: TST.PAT,*	

```
5735                                     :ML4AD
5736                                     :
5737                                     ROUTINE DECLARATION SECTION
5738 023520 016102 015412                MOV    ML.REG+6(R1),R2
5739 023524 050002                       BIS    R0,R2
5740 023526 010277 172102                MOV    R2,@ML.REG+230
5741 023532 152777 000020 171764        BISB   #20,@ML.REG+120
5742 023540 142777 000100 171756        BICB   #100,@ML.REG+120
5743 023546 142777 000010 171750        BICB   #10,@ML.REG+120
5744 023554 000207                       RTS    PC
5745
5746                                     : Routine Size: 35 words
5747                                     : Maximum stack depth per invocation: 3 words
5752
5753
```

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

3658
3659
3660
3629

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (33)

5755 :ML4AD
5756 :
5757 :
5758 :
5759 :
5760 :
5761 :
5762 :
5763 :
5764 :
5765 :
5766 :
5767 :
5768 :
5769 :
5770 :
5771 :
5772 :
5773 :
5774 :
5775 :
5776 :
5777 :
5778 :
5779 :
5780 :
5781 :
5782 :
5783 :
5784 :
5785 :
5786 :
5787 :
5788 :
5789 :
5790 :
5791 :
5792 :
5793 :
5794 :
5795 :
5796 :
5797 :
5798 :
5799 :
5800 :
5801 :
5802 :
5803 :
5804 :
5805 :
5806 :
5807 :
5808 :
5809 :

3662
3663
3664
3665
3666
3667
3668
3669
3670
3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713

ROUTINE DECLARATION SECTION
routine RD_PD (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
PROM DATA 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

--

5811 :ML4AD

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (33)

```

5814 : 3714
5815 : 3715 .ERR_FLG = ZERO; !CLEAR ERROR FLG
5816 : 3716 PROM_RW = ONE; !SET PROM READ WRITE
5817 : 3717 DAT_DM = ONE; !SET DATA DIAG MODE
5818 : 3718 WT_DATA = WRT_MASK; !SAVE THE DATA WRITTEN TO THE REGISTER
5819 : 3719 RD_DATA = .MLPD or .IGNORE; !READ AND SAVE THE REGISTER
5820 : 3720
5821 : 3721 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE; !READ THE REG FOR WRT_MASK
5822 : 3722
5823 : 3723
5824 : 3724 PROM_RW = ZERO; !SET ERROR FLAG IF NEQ
5825 : 3725 DAT_DM = ZERO; !CLEAR PROM READ WRTE
5826 : 3726 end; !CLEAR DATA DIAG MODE
5830 :
5831 :
  
```

```

5835 023556 004167 160756 RD.PD: .SBTTL RD.PD ROUTINE DECLARATION SECTION
5836 023562 005076 000010 JSR R1,$SAVE2
5837 023566 152777 000100 171730 CLR @10(SP) ; ERR.FLG
5838 023574 152777 000010 171722 BISB #100,@ML.REG+120 ;
5839 023602 016600 000012 MOV #10,@ML.REG+120 ;
5840 023606 006300 ASL 12(SP),R0 ; INDEX,*
5841 023610 006300 ASL R0
5842 023612 006300 ASL R0
5843 023614 010001 MOV R0,R1
5844 023616 016100 015406 MOV ML.REG+2(R1),R0
5845 023622 056600 000014 BIS 14(SP),R0 ; TST.PAT,*
5846 023626 046100 015410 BIC ML.REG+4(R1),R0
5847 023632 016102 015412 MOV ML.REG+6(R1),R2
5848 023636 050002 BIS R0,R2
5849 023640 010267 171504 MOV R2,WT_DATA
5850 023644 017702 171764 MOV @ML.REG+230,R2
5851 023650 056102 015412 BIS ML.REG+6(R1),R2 ;
5852 023654 010267 171472 MOV R2,RD_DATA ;
5853 023660 026767 171464 171464 CMP WT_DATA,RD_DATA ;
5854 023666 001403 BEQ 1$ ;
5855 023670 012776 000001 000010 MOV #1,@10(SP) ; *.ERR.FLG
5856 023676 142777 000100 171620 1$: BICB #100,@ML.REG+120 ;
5857 023704 142777 000010 171612 BICB #10,@ML.REG+120 ;
5858 023712 000207 RTS PC ;
5859 :
5860 :
5861 :
  
```

: Routine Size: 47 words
 : Maximum stack depth per invocation: 3 words

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (34)

5867 :ML4AD
5868 :
5869 :
5870 :
5871 :
5872 :
5873 :
5874 :
5875 :
5876 :
5877 :
5878 :
5879 :
5880 :
5881 :
5882 :
5883 :
5884 :
5885 :
5886 :
5887 :
5888 :
5889 :
5890 :
5891 :
5892 :
5893 :
5894 :
5895 :
5899 :
5900 :
5904 023714 000207
5905 :
5906 :
5907 :
5912 :
5913 :

ROUTINE DECLARATION SECTION

3727 routine WRT_EL (TST_PAT, index) : novalue =
3728 begin

3730 ++

FUNCTIONAL DESCRIPTION:

DUMMY ROUTINE CALL TO ASSIST IN THE READ
WRITE REGISTER ALGORITHM

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 REGISTERS ADDRESS.
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

3747 --

3748 :
3749 :
3750 : ERROR LOCATION REG IS READ ONLY
3751 return;
3752 end;

.SBTTL WRT.EL ROUTINE DECLARATION SECTION
WRT.EL: RTS PC :

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

3727

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (35)

5915 :ML4AD
5916 :
5917 :
5918 :
5919 :
5920 :
5921 :
5922 :
5923 :
5924 :
5925 :
5926 :
5927 :
5928 :
5929 :
5930 :
5931 :
5932 :
5933 :
5934 :
5935 :
5936 :
5937 :
5938 :
5939 :
5940 :
5941 :
5942 :
5943 :
5944 :
5945 :
5946 :
5947 :
5948 :
5949 :
5950 :
5951 :
5952 :
5953 :
5954 :
5955 :
5956 :
5957 :
5958 :
5959 :
5960 :
5961 :
5962 :
5963 :
5964 :
5965 :
5966 :
5967 :
5968 :
5969 :

3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804

ROUTINE DECLARATION SECTION
routine RD_EL (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
ERROR LOCATION REGISTER
WITH THE CONTENTS OF TST_PAT.

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;

5971 ;ML4AD

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (35)

5972 :
5973 :
5974 : 3805
5975 : 3806
5976 : 3807
5977 : 3808
5978 : 3809
5979 : 3810

```

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

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

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

5983
5984
5988
5989
5990
5991
5992
5993
5994
5995
5996
5997
6002
6003

023716	005076	000002	
023722	016667	000006	171420
023730	017767	171670	171414
023736	026767	171410	171404
023744	001403		
023746	012776	000001	000002
023754	000207		

.SBTTL RD.EL ROUTINE DECLARATION SECTION

```

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

3804
3805
3806
3808
3753

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

6005 :ML4AD
6006 :
6007 :
6008 :
6009 :
6010 :
6011 :
6012 :
6013 :
6014 :
6015 :
6016 :
6017 :
6018 :
6019 :
6020 :
6021 :
6022 :
6023 :
6024 :
6025 :
6026 :
6027 :
6028 :
6029 :
6030 :
6031 :
6032 :
6036 :
6037 :
6041 023756 000207
6042 :
6043 :
6044 :
6049 :
6050 :

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (36)

3811 routine WRT_EE (TST_PAT, index) : novalue =
3812 begin

!++

FUNCTIONAL DESCRIPTION:

DUMMY ROUTINE CALL TO ASSIST IN THE READ
WRITE REGISTER ALGORITHM

FORMAL PARAMETERS:

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

INDEX

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

!--

! ECC ERROR REGISTER IS READ ONLY
return;
end;

.SBTTL WRT.EE ROUTINE DECLARATION SECTION
WRT.EE: RTS PC ;

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

3811

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (37)

6052 :ML4AD
6053 :
6054 :
6055 :
6056 :
6057 :
6058 :
6059 :
6060 :
6061 :
6062 :
6063 :
6064 :
6065 :
6066 :
6067 :
6068 :
6069 :
6070 :
6071 :
6072 :
6073 :
6074 :
6075 :
6076 :
6077 :
6078 :
6079 :
6080 :
6081 :
6082 :
6083 :
6084 :
6085 :
6086 :
6087 :
6088 :
6089 :
6090 :
6091 :
6092 :
6093 :
6094 :
6095 :
6096 :
6097 :
6098 :
6099 :
6100 :
6101 :
6102 :
6103 :
6104 :
6105 :
6106 :

ROUTINE DECLARATION SECTION

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

++

FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
ECC ERROR REGISTER WITH TST_PAT.

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
COMPARED AGAINST THE CONTENTS
OF THE REGISTER UNDER TEST.

INDEX
USED BY THE MACRO MLEE TO
SELECT THE CURRENT REGISTERS ADDRESS.

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 TST_PAT
THUS ALLOWING CALLER TO PRINT
THE FAILING 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;
WT_DATA = .TST_PAT;
RD_DATA = .MLEE;

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

3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887

6108 :ML4AD

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS-20 BLISS-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (37)

```

6109 :
6110 :
6111 :      3888      if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;      !READ MLEE FOR TST_PAT
6112 :      3889
6113 :      3890
6114 :      3891      end;
6118 :
6119 :

```

!SET THE ERROR FLAG IF NEG

6123	023760	005076	000002		RD.EE:	.SBTTL	RD.EE ROUTINE DECLARATION SECTION		
6124	023764	016667	000006	171356		CLR	@2(SP)	:	ERR.FLG
6125	023772	017767	171616	171352		MOV	6(SP),WT.DATA	:	TST.PAT,*
6126	024000	026767	171346	171342		MOV	@ML.REG+210,RD.DATA	:	
6127	024006	001403				CMF	RD.DATA,WT.DATA	:	
6128	024010	012776	000001	000002		BEQ	1\$:	
6129	024016	000207			1\$:	MOV	#1,@2(SP)	:	*.ERR.FLG
6130						RTS	PC	:	
								:	

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

6131
6132
6137
6138

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (38)

6140 ;ML4AD
 6141 :
 6142 :
 6143 :
 6144 :
 6145 :
 6146 :
 6147 :
 6148 :
 6149 :
 6150 :
 6151 :
 6152 :
 6153 :
 6154 :
 6155 :
 6156 :
 6157 :
 6158 :
 6159 :
 6160 :
 6164 :
 6165 :
 6169 024020 004167 160514
 6170 024024 152777 000010 171472
 6171 024032 016600 000010
 6172 024036 006300
 6173 024040 006300
 6174 024042 006300
 6175 024044 010001
 6176 024046 016100 015406
 6177 024052 056600 000012
 6178 024056 046100 015410
 6179 024062 016102 015412
 6180 024066 050002
 6181 024070 010277 171500
 6182 024074 142777 000010 171422
 6183 024102 000207
 6184 :
 6185 :
 6186 :
 6191 :
 6192 :

ROUTINE DECLARATION SECTION

routine WRT_D1 (TST_PAT, index) : novalue =
 begin

```

  !++
  FUNCTIONAL DESCRIPTION:
  LOADS THE DATA DIAG 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.
  !--
  
```

```

  DAT_DM = ONE;           !SET DATA DIAG MODE
  MLD1 = WRT_MASK;       !LOAD MLD1 WITH GENERATED WRT_MASK PATTERN
  DAT_DM = ZERO;         !CLEAR DATA DIAG MODE
  end;
  
```

.SBTTL WRT.D1 ROUTINE DECLARATION SECTION

```

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
  
```

3892
 3906
 3907
 3908
 3892

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (39)

6194 :ML4AD
6195 :
6196 :
6197 :
6198 :
6199 :
6200 :
6201 :
6202 :
6203 :
6204 :
6205 :
6206 :
6207 :
6208 :
6209 :
6210 :
6211 :
6212 :
6213 :
6214 :
6215 :
6216 :
6217 :
6218 :
6219 :
6220 :
6221 :
6222 :
6223 :
6224 :
6225 :
6226 :
6227 :
6228 :
6229 :
6230 :
6231 :
6232 :
6233 :
6234 :
6235 :
6236 :
6237 :
6238 :
6239 :
6240 :
6241 :
6242 :
6243 :
6244 :
6245 :
6246 :
6247 :
6248 :

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
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961

ROUTINE DECLARATION SECTION

routine RD_D1 (TST_PAT, index, ERR_FLG) : novalue =
begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
DATA DIAG 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

--

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
27-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (39)

```

6250 :ML4AD
6251 :
6252 :
6253 : 3962
6254 : 3963 .ERR_FLG = ZERO;
6255 : 3964 DAT_DM = ONE;
6256 : 3965 ML_FUNC = write;
6257 : 3966 DAT_CLK = ONE;
6258 : 3967
6259 : 3968 if .REG_INIT_FLG IS_SET
6260 : 3969 then
6261 : 3970 begin
6262 : 3971 CLR_MBUS;
6263 : 3972 DAT_DM = ONE;
6264 : 3973 end;
6265 : 3974
6266 : 3975 WT_DATA = WRT_MASK;
6267 : 3976 RD_DATA = .MLD1;
6268 : 3977
6269 : 3978 if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;
6270 : 3979
6271 : 3980 CLR_MBUS;
6272 : 3981 end;
6273 : 3982
6274 : 3983
6278 :
6279 :

```

```

!CLEAR THE ERROR FLG
!SET DATA DIAG MODE
!LOAD MLCST WITH WRITE FUNCTION
!DO A DATA CLK

!SEE IF CALLER IS REG INIT TEST
!SET ERROR FLAG IF NEQ

!CLEAR MBUS TO GENERATE INIT DATA

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

!READ MLD1 FOR WRT_MASK

!SET ERR FLG IF NEQ
!CLEAR THE MBUS

```

```

6283 024104 004167 160430
6284 024110 005076 000010
6285 024114 152777 000010 171402
6286 024122 112777 000077 171254
6287 024130 152777 000061 171246
6288 024136 152777 000020 171360
6289 024144 026727 171210 000001
6290 024152 001017
6291 024154 152777 000040 171262
6292 024162 016701 171644
6293 024166 042701 177770
6294 024172 142777 000007 171244
6295 024200 150177 171240
6296 024204 152777 000010 171312
6297 024212 016600 000012
6298 024216 006300
6299 024220 006300
6300 024222 006300
6301 024224 010001
6302 024226 016100 015406
6303 024232 056600 000014
6304 024236 046100 015410

```

```

RD.D1: .SBTTL RD.D1 ROUTINE DECLARATION SECTION
JSR R1,$SAVE2
CLR @10(SP)
BISB #10,@ML.REG+120
BICB #77,@ML.REG
BISB #61,@ML.REG
BISB #20,@ML.REG+120
CMP REG.INIT.FLG,#1
SNE 1$
BISB #40,@ML.REG+40
MOV ML.DUT,R1
BIC #177770,R1
BICB #7,@ML.REG+40
BISB R1,@ML.REG+40
BISB #10,@ML.REG+120
1$: MOV 12(SP),R0
ASL R0
ASL R0
ASL R0
MOV R0,R1
MOV ML.REG+2(R1),R0
BIS 14(SP),R0
BIC ML.REG+4(R1),R0

```

```

3910
3963
3964
3965
3966
3968
3971
3973
3976

```

```

: ERR.FLG
:
:
:
:
:
: INDEX,*
: TST.PAT,*

```



```

6306          ;ML4AD
6307          ;
6308          ;
6309 024242 016102 015412          MOV ML.REG+6(R1),R2
6310 024246 050002          BIS  R0,R2
6311 024250 010267 171074          MOV  R2,WT.DATA
6312 024254 017767 171574 171070  MOV @ML.REG+170,RD.DATA
6313 024262 026767 171064 171060  CMP  RD.DATA,WT.DATA
6314 024270 001403          BEQ  2$
6315 024272 012776 000001 000010  MOV  #1,@10(SP)
6316 024300 152777 000040 171136 2$: BISB #40,@ML.REG+40
6317 024306 016702 171520          MOV  ML.DUT,R2
6318 024312 042702 177770          BIC  #177770,R2
6319 024316 142777 000007 171120  BICB #7,@ML.REG+40
6320 024324 150277 171114          BISB R2,@ML.REG+40
6321 024330 000207          RTS  PC
6322
6323          ; Routine Size: 75 words
6324          ; Maximum stack depth per invocation: 3 words
6329
6330

```

```

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<
3977
3979
3910

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (40)

6332 :ML4AD
 6333 :
 6334 :
 6335 :
 6336 :
 6337 :
 6338 :
 6339 :
 6340 :
 6341 :
 6342 :
 6343 :
 6344 :
 6345 :
 6346 :
 6347 :
 6348 :
 6349 :
 6350 :
 6351 :
 6352 :
 6356 :
 6357 :
 6361 :
 6362 :
 6363 :
 6364 :
 6365 :
 6366 :
 6367 :
 6368 :
 6369 :
 6370 :
 6371 :
 6372 :
 6373 :
 6374 :
 6375 :
 6376 :
 6377 :
 6378 :
 6383 :
 6384 :

3984
 3985
 3986
 3987
 3988
 3989
 3990
 3991
 3992
 3993
 3994
 3995
 3996
 3997
 3998
 3999
 4000
 4001

ROUTINE DECLARATION SECTION

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 MLD2 WITH GENERATED WRT_MASK PATTERN
 !CLEAR DATA DIAG MODE

.SBTTL WRT.D2 ROUTINE DECLARATION SECTION

```

WRT.D2: 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+200
        BICB  #10,@ML.REG+120
        RTS   PC
  
```

3984
 3998
 3999
 4000
 3984

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

29-Mar-1982 15:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (41)

6386 :ML4AD
6387 :
6388 :
6389 : 4002
6390 : 4003
6391 : 4004
6392 : 4005
6393 : 4006
6394 : 4007
6395 : 4008
6396 : 4009
6397 : 4010
6398 : 4011
6399 : 4012
6400 : 4013
6401 : 4014
6402 : 4015
6403 : 4016
6404 : 4017
6405 : 4018
6406 : 4019
6407 : 4020
6408 : 4021
6409 : 4022
6410 : 4023
6411 : 4024
6412 : 4025
6413 : 4026
6414 : 4027
6415 : 4028
6416 : 4029
6417 : 4030
6418 : 4031
6419 : 4032
6420 : 4033
6421 : 4034
6422 : 4035
6423 : 4036
6424 : 4037
6425 : 4038
6426 : 4039
6427 : 4040
6428 : 4041
6429 : 4042
6430 : 4043
6431 : 4044
6432 : 4045
6433 : 4046
6434 : 4047
6435 : 4048
6436 : 4049
6437 : 4050
6438 : 4051
6439 : 4052
6440 : 4053

ROUTINE DECLARATION SECTION

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 REGISTERS 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

--

```

6442 :ML4AD
6443 :
6444 :
6445 : 4054 .ERR_FLG = ZERO;
6446 : 4055 DAT_DM = ONE;
6447 : 4056 ML_FUNC = write;
6448 : 4057 DAT_CLK = ONE;
6449 : 4058
6450 : 4059 if .REG_INIT_FLG IS_SET
6451 : 4060 then
6452 : 4061 begin
6453 : 4062 CLR_MBUS;
6454 : 4063 DAT_DM = ONE;
6455 : 4064 end;
6456 : 4065
6457 : 4066 WT_DATA = WRT_MASK;
6458 : 4067 RD_DATA = .MLD2;
6459 : 4068
6460 : 4069 if .RD_DATA neq .WT_DATA then .ERR_FLG = ONE;
6461 : 4070
6462 : 4071
6463 : 4072 CLR_MBUS;
6464 : 4073 end;

```

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (41)
!CLEAR ERROR FLAG
!SET DATA DIAG MODE
!LOAD WRITE FUNC TO CS1
!DO A DATA CLOCK
!SEE IF CALLER IS REG INIT TEST
!CLEAR MBUS TO GENERATE INIT DATA
!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER
!READ MLD2 FR WRT_MASK PATTERN
!SET ERROR FLAG IF NEQ
!CLR MASS BUSS

```

Address	Hex	Hex	Hex	Label	Instruction	Comment	Address
6473	024416	004167	160116	RD.D2:	.SBTTL JSR R1,\$SAVE2		4002
6474	024422	005076	000010		CLR @10(SP)	ERR.FLG	4054
6475	024426	152777	000010	171070	BISB #10,@ML.REG+120		4055
6476	024434	142777	000077	170742	BICB #77,@ML.REG		4056
6477	024442	152777	000061	170734	BISB #61,@ML.REG		
6478	024450	152777	000020	171046	BISB #20,@ML.REG+120		
6479	024456	026727	170676	000001	CMP REG_INIT_FLG,#1		4057
6480	024464	001017			BNE 1\$		4059
6481	024466	152777	000040	170750	BISB #40,@ML.REG+40		
6482	024474	016701	171332		MOV ML,DUT,R1		4061
6483	024500	042701	177770		BIC #177770,R1		
6484	024504	142777	000007	170732	BICB #7,@ML.REG+40		
6485	024512	150177	170726		BISB R1,@ML.REG+40		
6486	024516	152777	000010	171000	BISB #10,@ML.REG+120		
6487	024524	016600	000012	1\$:	MOV 12(SP),R0	INDEX,*	4063
6488	024530	006300			ASL R0		4066
6489	024532	006300			ASL R0		
6490	024534	006300			ASL R0		
6491	024536	010001			MOV R0,R1		
6492	024540	016100	015406		MOV ML,REG+2(R1),R0		
6493	024544	056600	000014		BIS 14(SP),R0	TST.PAT,*	
6494	024550	046100	015410		BIC ML,REG+4(R1),R0		
6495	024554	016102	015412		MOV ML,REG+6(R1),R2		
6496	024560	050002			BIS R0,R2		

SEQ 0157

```

6498
6499
6500
6501 024562 010267 170562
6502 024566 017767 171012 170556
6503 024574 026767 170552 170546
6504 024602 001403
6505 024604 012776 000001 000010
6506 024612 152777 000040 170624 2$:
6507 024620 016702 171206
6508 024624 042702 177770
6509 024630 142777 000007 170606
6510 024636 150277 170602
6511 024642 000207
6512
6513
6514
6519
6520
  
```

:ML4AD
:

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

```

MOV R2,WT.DATA
MOV @ML.REG+200,RD.DATA
CMP RD.DATA,WT.DATA
BEQ 2$
MOV #1,@10(SP)
BISB #40,@ML.REG+40
MOV ML.DUT,R2
BIC #177770,R2
BICB #7,@ML.REG+40
BISB R2,@ML.REG+40
RTS PC
  
```

4067
4069

: *.ERR.FLG

4002

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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (42)

```

6522 :ML4AD
6523 :
6524 :
6525 : 4074 routine WRT_D3 (TST_PAT, index) : novalue =
6526 : 4075 begin
6527 : 4076
6528 : 4077
6529 : 4078
6530 : 4079
6531 : 4080
6532 : 4081
6533 : 4082
6534 : 4083
6535 : 4084
6536 : 4085
6537 : 4086
6538 : 4087
6539 : 4088
6540 : 4089
6541 : 4090
6542 : 4091
6543 : 4092
6544 : 4093
6545 : 4094
6546 : 4095
6547 : 4096
6548 : 4097
6549 : 4098
6550 : 4099
6551 : 4100
6552 : 4101
6553 : 4102
6554 : 4103
6555 : 4104
6556 : 4105
  
```

ROUTINE DECLARATION SECTION
 routine WRT_D3 (TST_PAT, index) : novalue =
 begin

```

!++
FUNCTIONAL DESCRIPTION:

LOADS THE DATA DIAG
REGISTER 3 WITH A DATA PATTERN
GENERATED BY THE MACRO
WRT_MASK.

FORMAL PARAMETERS:

TST PAT
CURRENT DATA PATTERN TO BE
LOADED IN THE REGISTER.

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

```

DAT_DM = ONE;
MLE2_MASK = %0'000377';
MLE2 = WRT_MASK;
MLE2_MASK = %0'100300';
DAT_DM = ZERO;
end;
  
```

```

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

```

6561 :
6565 024644 004167 157670 WRT.D3: .SBTTL WRT.D3 ROUTINE DECLARATION SECTION
6566 024650 152777 000010 170646 JSP R1,$SAVE2
6567 024656 012767 000377 170706 BISB #10,ML.REG+120
6568 024664 016600 000010 MOV #3,ML.REG+166
6569 024670 006300 MOV 10(SP),R0
6570 024672 006300 ASL R0
6571 024674 006300 ASL R0
6572 024676 010001 ASL R0
6573 024700 016100 015406 MOV R0,R1
6574 024704 056600 000012 MOV ML.REG+2(R1),R0
6575 024710 046100 015410 BIS 12(SP),R0
6576 024714 016102 015412 BIC ML.REG+4(R1),R0
MOV ML.REG+6(R1),R2
  
```

4074
 4100
 4101
 4102

: INDEX,*
 : TST.PAT,*

```
6578                                     :ML4AD
6579                                     :
6580                                     ROUTINE DECLARATION SECTION
6581 024720 050002                       BIS      R0,R2
6582 024722 010277 170636                 MOV      R2,@ML.REG+160
6583 024726 012767 100300 170636         MOV      #-77500,ML.REG+166
6584 024734 142777 000010 170562         BICB    #10,@ML.REG+120
6585 024742 000207                       RTS      PC
6586                                     :
6587                                     : Routine Size: 32 words
6588                                     : Maximum stack depth per invocation: 3 words
6593
6594
```

```
29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<
4103
4104
4074
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (43)

6596 :ML4AD
6597 :
6598 :
6599 : 4106
6600 : 4107
6601 : 4108
6602 : 4109
6603 : 4110
6604 : 4111
6605 : 4112
6606 : 4113
6607 : 4114
6608 : 4115
6609 : 4116
6610 : 4117
6611 : 4118
6612 : 4119
6613 : 4120
6614 : 4121
6615 : 4122
6616 : 4123
6617 : 4124
6618 : 4125
6619 : 4126
6620 : 4127
6621 : 4128
6622 : 4129
6623 : 4130
6624 : 4131
6625 : 4132
6626 : 4133
6627 : 4134
6628 : 4135
6629 : 4136
6630 : 4137
6631 : 4138
6632 : 4139
6633 : 4140
6634 : 4141
6635 : 4142
6636 : 4143
6637 : 4144
6638 : 4145
6639 : 4146
6640 : 4147
6641 : 4148
6642 : 4149
6643 : 4150
6644 : 4151
6645 : 4152
6646 : 4153
6647 : 4154
6648 : 4155
6649 : 4156
6650 : 4157

ROUTINE DECLARATION SECTION
routine RD_D3 (TST_PAT, index, FRR_FLG) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
DATA DIAG REGISTER 3
WITH THE MASKED DATA PATTERN
GENERATED BY THE MACRO 'WRT_MASK'.

IF THE COMPARE IS NOT EQUAL 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

--

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (43)

```

6652 :ML4AD
6653 :
6654 :
6655 : 4158
6656 : 4159 .ERR_FLG = ZERO;
6657 : 4160 MLE2_MASK = %0'000377';
6658 : 4161 DAT_DM = ONE;
6659 : 4162 ML_FUNC = write;
6660 : 4163 DAT_CLK = ONE;
6661 : 4164
6662 : 4165 if .REG_INIT_FLG IS_SET
6663 : 4166 then
6664 : 4167 begin
6665 : 4168 CLR_MBUS;
6666 : 4169 DAT_DM = ONE;
6667 : 4170 end;
6668 : 4171
6669 : 4172 WT_DATA = WRT_MASK;
6670 : 4173 RD_DATA = .MLE2 or .IGNORE;
6671 : 4174
6672 : 4175 if .WT_DATA neq .RD_DATA then .ERR_FLG = ONE;
6673 : 4176
6674 : 4177 MLE2_MASK = %0'100300';
6675 : 4178 CLR_MBUS;
6676 : 4179 end;
6680 :
6681 :

```

```

!CLEAR ERROR FLAG
!SET DATA DIAG MODE
!LOAD WRITE FUNCTION TO MLCS1
!DO A DATA CLOCK
!SEE IF CALLER IS REG INIT TEST
!CLEAR MBUS TO GENERATE INIT DATA
!SAVE THE DATA WRITTEN TO THE REGISTER
!READ AND SAVE THE REGISTER
!READ THE REG FOR WRT_MASK
!CLEAR THE MASS BUS

```

```

6685 024744 004167 157570 RD.D3: .SBTTL RD.D3 ROUTINE DECLARATION SECTION
6686 024750 005076 000010 JSR R1,SSAVE2
6687 024754 012767 000377 170610 CLR @10(SP)
6688 024762 152777 000010 170534 MOV #377,ML.REG+166
6689 024770 142777 000077 170406 BICB #10,2ML.REG+120
6690 024776 152777 000061 170400 BICB #77,2ML.REG
6691 025004 152777 000020 170512 BICB #61,2ML.REG
6692 025012 026727 170342 000001 BICB #20,2ML.REG+120
6693 025020 001017 CMP REG.INIT.FLG,#1
6694 025022 152777 000040 170414 BNE 18
6695 025030 016701 170776 BICB #40,2ML.REG+40
6696 025034 042701 177770 MOV ML,DUT,R1
6697 025040 142777 000007 170376 BIC #177770,R1
6698 025046 150177 170372 BICB #7,2ML.REG+40
6699 025052 152777 000010 170444 BICB R1,2ML.REG+40
6700 025060 016600 000012 18 BICB #10,2ML.REG+120
6701 025064 006300 18: MOV 12(SP),R0
6702 025066 006300 ASL R0
6703 025070 006300 ASL R0
6704 025072 010001 ASL R0
6705 025074 016100 015406 MOV R0,R1
6706 025100 056600 000014 MOV ML.REG+2(R1),R0
        BIS 14(SP),R0

```

```

: ERR.FLG 4106
: 4159
: 4160
: 4161
: 4162
: 4163
: 4165
: 4167
: 4169
: INDEX,* 4169
: 4172
: TST.PAT,*

```

```

6708          :ML4AD
6709          :
6710          :
6711 025104 046100 015410          BIC    ML.REG+4(R1),R0
6712 025110 016102 015412          MOV    ML.REG+6(R1),R2
6713 025114 050002                   BIS    R0,R2
6714 025116 010267 170226          MOV    R2,WT.DATA
6715 025122 017702 170436          MOV    @ML.REG+160,R2
6716 025126 056102 015412          BIS    ML.REG+6(R1),R2
6717 025132 010267 170214          MOV    R2,RD.DATA
6718 025136 026767 170206          CMP    WT.DATA,RD.DATA
6719 025144 001403                   BEQ    2$
6720 025146 012776 000001 000010  2$:   MOV    #1,@10(SP)
6721 025154 012767 100300 170410  :   MOV    #-77500,ML.REG+166
6722 025162 152777 000040 170254  :   BISB  #40,@ML.REG+40
6723 025170 016702 170636          MOV    ML.DUT,R2
6724 025174 042702 177770          BIC    #177770,R2
6725 025200 142777 000007 170236  :   BICB  #7,@ML.REG+40
6726 025206 150277 170232          BISB  R2,@ML.REG+40
6727 025212 000207                   RTS    PC
6728          :
6729          :
6730          :
6735          :
6736          :
  
```

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

4173

4175

4177

4106

: Routine Size: 84 words
 : Maximum stack depth per invocation: 3 words

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (44)

6738 :ML4AD
6739 :
6740 :
6741 :
6742 :
6743 :
6744 :
6745 :
6746 :
6747 :
6748 :
6749 :
6750 :
6751 :
6752 :
6753 :
6754 :
6755 :
6756 :
6757 :
6758 :
6759 :
6760 :
6761 :
6762 :
6763 :
6764 :
6765 :
6766 :
6770 :
6771 :
6775 025214 000207
6776 :
6777 :
6778 :
6783 :
6784 :

ROUTINE DECLARATION SECTION

routine WRT_DS (TST_PAT, index) : novalue =
begin

!++

FUNCTIONAL DESCRIPTION:

DUMMY ROUTINE CALL TO ASSIST IN THE READ
WRITE REGISTER ALGORITHM

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 REGISTERS ADDRESS,
FORCED HI, FORCED LO AND DON'T CARE
MASK INFORMATION.

!--

DRIVE STATUS REG IS READ ONLY
return;
end;

.SBTTL WRT_DS ROUTINE DECLARATION SECTION
WRT_DS: RTS PC :

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (45)

6786 :ML4AD
6787 :
6788 :
6789 :
6790 :
6791 :
6792 :
6793 :
6794 :
6795 :
6796 :
6797 :
6798 :
6799 :
6800 :
6801 :
6802 :
6803 :
6804 :
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 :

ROUTINE DECLARATION SECTION

4206 routine RD_DS (TST_PAT, index, ERR_FLG) : novalue =
4207 begin

++
FUNCTIONAL DESCRIPTION:

COMPARES THE CONTENTS OF THE
DRIVE STATUS 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

--

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (45)

6842 : ML4AD
6843 :
6844 :
6845 : 4258
6846 : 4259
6847 : 4260
6848 : 4261
6849 : 4262
6850 : 4263
6851 : 4264
6852 : 4265
6853 : 4266
6857 :
6858 :

ROUTINE DECLARATION SECTION

```

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

```

6862 025216 004167 157316
6863 025222 005076 000010
6864 025226 016600 000012
6865 025232 006300
6866 025234 006300
6867 025236 006300
6868 025240 010001
6869 025242 016100 015406
6870 025246 056600 000014
6871 025252 046100 015410
6872 025256 016102 015412
6873 025262 050002
6874 025264 010267 170060
6875 025270 017702 170160
6876 025274 056102 015412
6877 025300 010267 170046
6878 025304 026767 170040 170040
6879 025312 001403
6880 025314 012776 000001 000010
6881 025322 000207
6882
6883
6884
6889
6890

```

.SBTTL RD.DS ROUTINE DECLARATION SECTION
RD.DS: JSR   R1,SSAVE2
        CLR   @10(SP)
        MOV   12(SP),R0
        ASL   R0
        ASL   R0
        ASL   R0
        MOV   R0,R1
        MOV   ML.REG+2(R1),R0
        BIS   14(SP),R0
        BIC   ML.REG+4(R1),R0
        MOV   ML.REG+6(R1),R2
        BIS   R0,R2
        MOV   R2,WT.DATA
        MOV   @ML.REG+50,R2
        BIS   ML.REG+6(R1),R2
        MOV   R2,RD.DATA
        CMP   WT.DATA,RD.DATA
        BEQ   1$
1$:     MOV   #1,@10(SP)
        RTS   PC

```

4206
4259
4260

4261
4263

4206

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.6LI.4 (46)

6892 :ML4AD
6893 :
6894 :
6895 :
6896 :
6897 :
6898 :
6899 :
6900 :
6901 :
6902 :
6903 :
6904 :
6905 :
6906 :
6907 :
6908 :
6909 :
6910 :
6911 :
6912 :
6913 :
6914 :
6915 :
6916 :
6917 :
6918 :
6919 :
6920 :
6921 :
6922 :
6923 :
6924 :
6925 :
6926 :
6927 :
6928 :
6929 :
6930 :
6931 :
6932 :
6933 :
6934 :
6935 :
6936 :
6937 :
6938 :
6939 :
6940 :
6941 :
6942 :
6943 :
6944 :
6945 :
6946 :

ROUTINE DECLARATION SECTION

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

!++

FUNCTIONAL DESCRIPTION:

A ROUTINE TO SELECTIVELY
CALLED ROUTINES WHICH
WRITE TO INDIVIDUAL ML11
REGISTERS

FORMAL PARAMETERS:

REG_SEL
CASE SELECT EXPRESSION TO
SELECT THE WRITE REGISTER
ROUTINE TO CALLED

TST_PAT
DATA PATTERN WHICH THE SELECTED
REGISTER WILL BE TESTED AGAINST

INDEX
LOADED WITH THE ML_REG INDEX
SELECT NUMBER OF THE REGISTER
BEING TESTED

SIDE EFFECTS:

WHEN A WRITE REGISTER ROUTINE IS CALLED
THE VARIABLE 'INDEX' FROM THE CALLING
TEST IS LOADED WITH THE REGISTERS
ML_REG INDEX NUMBER.

THIS ENABLES THE CALLING TEST TO FIND
THE FAILING REGISTER ADDRESS.

!--

case .REG_SEL from 0 to 13 of
set

!SELECT THE WRITE REGISTER ROUTINE CALL

[0] :
WRT_CS1 (.TST_PAT, .index = 0); !CALL ROUTINE TO LOAD MLCS1
[1] :
WRT_ER (.TST_PAT, .index = 6); !CALL ROUTINE TO LOAD MLER
[2] :
WRT_DA (.TST_PAT, .index = 3); !CALL ROUTINE TO LOAD MLDA
[3] :
WRT_MR (.TST_PAT, .index = 10); !CALL ROUTINE TO LOAD MLMR
[4] :

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (46)

6948 :ML4AD
 6949 :
 6950 :
 6951 :
 6952 :
 6953 :
 6954 :
 6955 :
 6956 :
 6957 :
 6958 :
 6959 :
 6960 :
 6961 :
 6962 :
 6963 :
 6964 :
 6965 :
 6966 :
 6967 :
 6968 :
 6969 :
 6970 :
 6971 :
 6972 :
 6973 :
 6974 :
 6975 :
 6976 :
 6977 :
 6978 :
 6979 :
 6980 :
 6981 :
 6985 :
 6986 :

4319
 4320
 4321
 4322
 4323
 4324
 4325
 4326
 4327
 4328
 4329
 4330
 4331
 4332
 4333
 4334
 4335
 4336
 4337
 4338
 4339
 4340
 4341
 4342
 4343
 4344
 4345
 4346
 4347
 4348
 4349

ROUTINE DECLARATION SECTION

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;

end;

025324 004167 157210
 025330 016600 000010
 025334 016601 000014
 025340 016602 000012
 025344 006302
 025346 066207 025352
 025352 000034
 025354 000050
 025356 000066
 025360 000104
 025362 000122
 025364 000140
 025366 000156

.SBTTL WRT.REG ROUTINE DECLARATION SECTION

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\$

4267
 4307
 4303

Address	Label	OpCode	OpData	OpComment	PC
7004					
7005					
7006					
7007	025370	000174			
7008	025372	000212			
7009	025374	000230			
7010	025376	000246			
7011	025400	000264			
7012	025402	000302			
7013	025404	000320			
7014	025406	010146			
7015	025410	005010			
7016	025412	005046			
7017	025414	004767	174272		
7018	025420	000532			
7019	025422	010146			
7020	025424	012710	000006		
7021	025430	011046			
7022	025432	004767	174432		
7023	025436	000523			
7024	025440	010146			
7025	025442	012710	000003		
7026	025446	011046			
7027	025450	004767	174572		
7028	025454	000514			
7029	025456	010146			
7030	025460	012710	000012		
7031	025464	011046			
7032	025466	004767	174732		
7033	025472	000505			
7034	025474	010146			
7035	025476	012710	000015		
7036	025502	011046			
7037	025504	004767	175300		
7038	025510	000476			
7039	025512	010146			
7040	025514	012710	000016		
7041	025520	011046			
7042	025522	004767	175470		
7043	025526	000467			
7044	025530	010146			
7045	025532	012710	000010		
7046	025536	011046			
7047	025540	004767	175036		
7048	025544	000460			
7049	025546	010146			
7050	025550	012710	000023		
7051	025554	011046			
7052	025556	004767	175666		
7053	025562	000451			
7054	025564	010146			
7055	025566	012710	000021		
7056	025572	011046			
7057	025574	004767	176156		
7058	025600	000442			

:ML4AD

ROUTINE DECLARATION SECTION

2\$:

3\$:

4\$:

5\$:

6\$:

7\$:

8\$:

9\$:

10\$:

```

.WORD 9$-1$
.WORD 10$-1$
.WORD 11$-1$
.WORD 12$-1$
.WORD 13$-1$
.WORD 14$-1$
.WORD 15$-1$
MOV R1, -(SP)
CLR (R0)
CLR -(SP)
JSR PC, WRT.CS1
BR 16$
MOV R1, -(SP)
MOV #6, (R0)
MOV (R0), -(SP)
JSR PC, WRT.ER
BR 16$
MOV R1, -(SP)
MOV #3, (R0)
MOV (R0), -(SP)
JSR PC, WRT.DA
BR 16$
MOV R1, -(SP)
MOV #12, (R0)
MOV (R0), -(SP)
JSR PC, WRT.MR
BR 16$
MOV R1, -(SP)
MOV #15, (R0)
MOV (R0), -(SP)
JSR PC, WRT.E1
BR 16$
MOV R1, -(SP)
MOV #16, (R0)
MOV (R0), -(SP)
JSR PC, WRT.E2
BR 16$
MOV R1, -(SP)
MOV #10, (R0)
MOV (R0), -(SP)
JSR PC, WRT.PA
BR 16$
MOV R1, -(SP)
MOV #23, (R0)
MOV (R0), -(SP)
JSR PC, WRT.PD
BR 16$
MOV R1, -(SP)
MOV #21, (R0)
MOV (R0), -(SP)
JSR PC, WRT.EE
BR 16$

```

4307

4303
4310

4303
4313

4303
4316

4303
4319

4303
4322

4303
4325

4303
4328

4303
4331

4303

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (47)

7099 :ML4AD
7100 :
7101 :
7102 :
7103 :
7104 :
7105 :
7106 :
7107 :
7108 :
7109 :
7110 :
7111 :
7112 :
7113 :
7114 :
7115 :
7116 :
7117 :
7118 :
7119 :
7120 :
7121 :
7122 :
7123 :
7124 :
7125 :
7126 :
7127 :
7128 :
7129 :
7130 :
7131 :
7132 :
7133 :
7134 :
7135 :
7136 :
7137 :
7138 :
7139 :
7140 :
7141 :
7142 :
7143 :
7144 :
7145 :
7146 :
7147 :
7148 :
7149 :
7150 :
7151 :
7152 :
7153 :

ROUTINE DECLARATION SECTION

routine RD_REG (TST_PAT, REG_SEL, ERR_FLG) : novalue =
begin

!++

FUNCTIONAL DESCRIPTION:

A ROUTINE TO SELECTIVELY
CALLED ROUTINES WHICH
READ TO INDIVIDUAL ML11
REGISTERS.

FORMAL PARAMETERS:

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.

REG_SEL
CASE SELECT EXPRESSION TO
SELECT THE WRITE REGISTER
ROUTINE TO CALLED

TST_PAT
DATA PATTERN WHICH THE SELECTED
REGISTER WILL BE TESTED AGAINST

!--

```

case .REG_SEL from 0 to 13 of
set
[0] : RD_CS1 (.TST_PAT, 0, .ERR_FLG); !CALL ROUTINE TO READ MLCS1
[1] : RD_ER (.TST_PAT, 6, .ERR_FLG); !CALL ROUTINE TO READ MLER
[2] : RD_DA (.TST_PAT, 3, .ERR_FLG); !CALL ROUTINE TO READ MLDA
[3] : RD_MR (.TST_PAT, 10, .ERR_FLG); !CALL ROUTINE TO READ MLMR
[4] : RD_E1 (.TST_PAT, 13, .ERR_FLG); !CALL ROUTINE TO READ MLE1
[5] : RD_E2 (.TST_PAT, 14, .ERR_FLG); !CALL ROUTINE TO READ MLE2
[6] : RD_PA (.TST_PAT, 8, .ERR_FLG); !CALL ROUTINE TO READ MLPA
    
```

7155 :ML4AD
7156 :
7157 :
7158 : 4402
7159 : 4403
7160 : 4404
7161 : 4405
7162 : 4406
7163 : 4407
7164 : 4408
7165 : 4409
7166 : 4410
7167 : 4411
7168 : 4412
7169 : 4413
7170 : 4414
7171 : 4415
7172 : 4416
7173 : 4417
7174 : 4418
7175 : 4419
7176 : 4420
7177 : 4421
7178 : 4422
7179 : 4423
7180 : 4424

ROUTINE DECLARATION SECTION

```

[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:

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (47)

end;

7184 :
7185 :
7189 025712 004167 156622
7190 025716 016600 000010
7191 025722 016601 000014
7192 025726 016602 000012
7193 025732 006302
7194 025734 066207 025740
7195 025740 000034
7196 025742 000050
7197 025744 000066
7198 025746 000104
7199 025750 000122
7200 025752 000140
7201 025754 000156
7202 025756 000174
7203 025760 000212
7204 025762 000230
7205 025764 000246
7206 025766 000264
7207 025770 000302
7208 025772 000320
7209 025774 010146

```

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

```

4350
4382
4378

4382

Address	OpCode	Operand1	Operand2	Label	Instruction	Comment	Address
7211							
7212							
7213							
7214	025776	005046			CLR	-(SP)	
7215	026000	010046			MOV	R0, -(SP)	
7216	026002	004767	173754		JSR	PC, RD.CS1	
7217	026006	000532			BR	16\$	
7218	026010	010146		3\$:	MOV	R1, -(SP)	4378
7219	026012	012746	000006		MOV	#6, -(SP)	4385
7220	026016	010046			MOV	R0, -(SP)	
7221	026020	004767	174114		JSR	PC, RD.ER	
7222	026024	000523			BR	16\$	
7223	026026	010146		4\$:	MOV	R1, -(SP)	4378
7224	026030	012746	000003		MOV	#3, -(SP)	4388
7225	026034	010046			MOV	R0, -(SP)	
7226	026036	004767	174254		JSR	PC, RD.DA	
7227	026042	000514			BR	16\$	
7228	026044	010146		5\$:	MOV	R1, -(SP)	4378
7229	026046	012746	000012		MOV	#12, -(SP)	4391
7230	026052	010046			MOV	R0, -(SP)	
7231	026054	004767	174414		JSR	PC, RD.MR	
7232	026060	000505			BR	16\$	
7233	026062	010146		6\$:	MOV	R1, -(SP)	4378
7234	026064	012746	000015		MOV	#15, -(SP)	4394
7235	026070	010046			MOV	R0, -(SP)	
7236	026072	004767	174776		JSR	PC, RD.E1	
7237	026076	000476			BR	16\$	
7238	026100	010146		7\$:	MOV	R1, -(SP)	4378
7239	026102	012746	000016		MOV	#16, -(SP)	4397
7240	026106	010046			MOV	R0, -(SP)	
7241	026110	004767	175200		JSR	PC, RD.E2	
7242	026114	000467			BR	16\$	
7243	026116	010146		8\$:	MOV	R1, -(SP)	4378
7244	026120	012746	000010		MOV	#10, -(SP)	4400
7245	026124	010046			MOV	R0, -(SP)	
7246	026126	004767	174534		JSR	PC, RD.PA	
7247	026132	000460			BR	16\$	
7248	026134	010146		9\$:	MOV	R1, -(SP)	4378
7249	026136	012746	000023		MOV	#23, -(SP)	4403
7250	026142	010046			MOV	R0, -(SP)	
7251	026144	004767	175406		JSR	PC, RD.PD	
7252	026150	000451			BR	16\$	
7253	026152	010146		10\$:	MOV	R1, -(SP)	4378
7254	026154	012746	000021		MOV	#21, -(SP)	4406
7255	026160	010046			MOV	R0, -(SP)	
7256	026162	004767	175572		JSR	PC, RD.EE	
7257	026166	000442			BR	16\$	
7258	026170	010146		11\$:	MOV	R1, -(SP)	4378
7259	026172	012746	000022		MOV	#22, -(SP)	4409
7260	026176	010046			MOV	R0, -(SP)	
7261	026200	004767	175512		JSR	PC, RD.EL	
7262	026204	000433			BR	16\$	
7263	026206	010146		12\$:	MOV	R1, -(SP)	4378
7264	026210	012746	000005		MOV	#5, -(SP)	4412
7265	026214	010046			MOV	R0, -(SP)	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

```
7267      :ML4AD  
7268      :  
7269      :  
7270 026216 004767 176774      JSR  PC, RD.DS  
7271 026222 000424      BR   16$  
7272 026224 010146      13$: MOV  R1, -(SP)  ::  
7273 026226 012746 000017      MOV  #17, -(SP)  ::  
7274 026232 010046      MOV  R0, -(SP)  ::  
7275 026234 004767 175644      JSR  PC, RD.D1  
7276 026240 000415      BR   16$  
7277 026242 010146      14$: MOV  R1, -(SP)  ::  
7278 026244 012746 000020      MOV  #20, -(SP)  ::  
7279 026250 010046      MOV  R0, -(SP)  ::  
7280 026252 004767 176140      JSR  PC, RD.D2  
7281 026256 000406      BR   16$  
7282 026260 010146      15$: MOV  R1, -(SP)  ::  
7283 026262 012746 000016      MOV  #16, -(SP)  ::  
7284 026266 010046      MOV  R0, -(SP)  ::  
7285 026270 004767 176450      JSR  PC, RD.D3  
7286 026274 062706 000006      16$: ADD  #6, SP  ::  
7287 026300 000207      RTS  PC  ::  
7288  
7289  
7290  
7295  
7296  
7297 :
```

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

4425 !<BLF/PAGE>

```

7299 :ML4AD
7300 :
7301 :
7302 : 4426
7303 : 4427 !
7304 : 4428 BGNMSG (DUMPER):
7308
7309
7313 : 302 004767 000004
7314 026306 104423
7315 026310 000207
7316
7317 : Routine Size: 4 words
7318 : Maximum stack depth per invocation: 0 words
7323
7324
7325 : 4429
7326 : 4430 !++
7327 : 4431 FUNCTIONAL DESCRIPTION
7328 : 4432 UPON COMPLETION OF ERROR MESSAGES
7329 : 4433 DUMP OUT ALL PERTINENT DRIVE AND
7330 : 4434 RH REGISTERS
7331 : 4435 !--
7332 : 4436
7333 : 4437 if .REGDMP
7334 : 4438 then
7335 : 4439 begin
7336 : 4440 PRINTB (ONE_FMT, PWR_14);
7337 : 4441 PRINTB (FMT_23);
7338 : 4442 PRINTB (FMT_24, REG_1, MLCS1, .MLCS1);
7339 : 4443 PRINTB (FMT_24, REG_18, MLWC, .MLWC);
7340 : 4444 PRINTB (FMT_24, REG_19, MLBA, .MLBA);
7341 : 4445 PRINTB (FMT_24, REG_6, MLDA, .MLDA);
7342 : 4446 PRINTB (FMT_24, REG_17, MLCS2, .MLCS2);
7343 : 4447 PRINTB (FMT_24, REG_2, MLDS, .MLDS);
7344 : 4448 PRINTB (FMT_24, REG_3, MLER, .MLER);
7345 : 4449 PRINTB (FMT_24, REG_5, MLAS, .MLAS);
7346 : 4450 PRINTB (FMT_24, REG_4, MLMR, .MLMR);
7347 : 4451 PRINTB (FMT_24, REG_7, MLDT, .MLDT);
7348 : 4452 PRINTB (FMT_24, REG_9, MLSN, .MLSN);
7349 : 4453 PRINTB (FMT_24, REG_14, MLEE, .MLEE);
7350 : 4454 PRINTB (FMT_24, REG_15, MLEL, .MLEL);
7351 : 4455
7352 : 4456 if .LST_DUT_REG eql 21
7353 : 4457 then
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (48)

4428

!PRINT REGISTER DUMP MESSAGE
 !PRINT DUMPER COLUMN HEADINGS
 !PRINT OUT THE RH & ML11 REGISTER CONTENTS

!SEE IF THIS IS A RH70
 !IF YES THEN PRINT RH70 REGISTERS

7355 :ML4AD
 7356 :
 7357 :
 7358 :
 7359 :
 7360 :
 7361 :
 7362 :
 7363 :
 7364 :
 7365 :
 7369 :

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (48)

4458 begin
 4459 PRINTB (FMT_24, REG_20, MLBAE, .MLBAE);
 4460 PRINTB (FMT_24, REG_21, MLCS3, .MLCS3);
 4461 end;
 4462
 4463 end;
 4464
 4465 ENDMSG;

				.SBTTL	MSDUMPER ROUTINE DECLARATION SECTION	
7374	026312			MSDUMPER:	BIT #1,REGDMP	4437
7375	026312	032767	000001 154064		BNE 1\$	
7376	026320	001001			RTS PC	
7377	026322	000207		1\$:	MOV #PHR.14,-(SP)	4440
7378	026324	012746 012104			MOV #ONE.FMT,-(SP)	
7379	026330	012746 010240			MOV #2,-(SP)	
7380	026334	012746 000002			MOV SP,R0	: SP,*
7381	026340	010600			TRAP 14	
7382	026342	104414			MOV #FMT.23,(SP)	
7383	026344	012716 010072			MOV #1,-(SP)	4441
7384	026350	012746 000C01			MOV SP,R0	: SP,*
7385	026354	010600			TRAP 14	
7386	026356	104414			MOV @ML.REG,(SP)	
7387	026360	017716 167020			MOV ML.REG,-(SP)	4442
7388	026364	016746 167014			MOV #REG.1,-(SP)	
7389	026370	012746 012500			MOV #FMT.24,-(SP)	
7390	026374	012746 010142			MOV #4,-(SP)	
7391	026400	012746 000004			MOV SP,R0	: SP,*
7392	026404	010600			TRAP 14	
7393	026406	104414			MOV @ML.REG+10,(SP)	
7394	026410	017716 167000			MOV ML.REG+10,-(SP)	4443
7395	026414	016746 166774			MOV #REG.18,-(SP)	
7396	026420	012746 012652			MOV #FMT.24,-(SP)	
7397	026424	012746 010142			MOV #4,-(SP)	
7398	026430	012746 000004			MOV SP,R0	: SP,*
7399	026434	010600			TRAP 14	
7400	026436	104414			MOV @ML.REG+20,(SP)	
7401	026440	017716 166760			MOV ML.REG+20,-(SP)	4444
7402	026444	016746 166754			MOV #REG.19,-(SP)	
7403	026450	012746 012660			MOV #FMT.24,-(SP)	
7404	026454	012746 010142			MOV #4,-(SP)	
7405	026460	012746 000004			MOV SP,R0	: SP,*
7406	026464	010600			TRAP 14	
7407	026466	104414			MOV @ML.REG+30,(SP)	
7408	026470	017716 166740			MOV ML.REG+30,-(SP)	4445
7409	026474	016746 166734				

7516 :ML4AD

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (49)

7519 : 4466 routine WRT_CHK_TRANSFER (SIZE, DST, SRC) : novalue =
 7520 : 4467 begin

7521 : 4468
 7522 : 4469 ++
 7523 : 4470 FUNCTIONAL DESCRIPTION:
 7524 : 4471 THIS ROUTINE WHEN CALLED WILL PERFORM
 7525 : 4472 A MASS BUS WRITE CHECK TRANSFER TO THE REQUESTED
 7526 : 4473 DESTINATION AND SOURCE ADDRESS OF LENGTH
 7527 : 4474 'SIZE'.
 7528 : 4475

7529 : 4476 FORMAL PARAMETERS:
 7530 : 4477 SIZE: DETERMINES THE NUMBER OF WORDS TO
 7531 : 4478 TRANSFER TO THE DESTINATION ADRS.
 7532 : 4479
 7533 : 4480 DST: DETERMINES THE DESTINATION ADRS OF
 7534 : 4481 TRANSFER.
 7535 : 4482
 7536 : 4483 SRC: DETERMINES THE SOURCE ADRS OF THE
 7537 : 4484 TRANSFER.
 7538 : 4485
 7539 : 4486 --

7540 : 4487	CLR_MBUS;	!CLEAR THE MASS BUS BEFORE WE START
7541 : 4488	ECC_DIS = ONE;	!DISABLE ERROR CORRECTION
7542 : 4489	BAI = ONE;	!MAKE THE TRANSFER STAY ON ONE BUS ADRS
7543 : 4490	MLWC = .SIZE;	!LOAD THE SIZE OF THE TRANSFER
7544 : 4491	MLDA = .DST;	!LOAD THE DESTINATION ADRS OF THE TRANSFER
7545 : 4492	MLBA = .SRC;	!LOAD THE SOURCE ADRS OF THE TRANSFER
7546 : 4493	MLCS1 = WRT_CHK;	!LOAD THE WRITE CHECK FUNCTION INTO CS1
7547 : 4494		
7548 : 4495		
7549 : 4496	if (.SC) and (not .WCE)	!DID LOADING THE FUNCTION CAUSE A SC ERROR
7550 : 4497	then	
7551 : 4498	begin	
7552 : 4499	ERRDF (148, INTER, DUMPER);	!REPORT THE ERROR TO OPERATOR
7553 : 4500	PRINTB (ONE FMT, WC_ERR);	!TELL WHAT THE ERROR IS
7554 : 4501	DODU (.ML_LDN);	!DROP THE UNIT
7555 : 4502	DOCLN;	!EXIT THE PROG
7556 : 4503	end;	
7557 : 4504		
7558 : 4505	do	!DO NOTHING
7559 : 4506	0	
7560 : 4507	until .DRY;	!UNTIL THE DRIVE IS READY
7561 : 4508		
7562 : 4509	if (.SC) and (not .WCE)	!DID THE SC BIT SET DURING THE XFER
7563 : 4510	then	
7564 : 4511	begin	
7565 : 4512	ERRDF (148, INTER, DUMPER);	!REPORT THE ERROR TO OPERATOR
7566 : 4513	PRINTB (ONE FMT, WC_ERR);	!TELL WHAT THE ERROR IS
7567 : 4514	DODU (.ML_LDN);	!DROP THE UNIT
7568 : 4515	DOCLN;	!EXIT THE PROGRAM
7569 : 4516	end;	
7570 : 4517		


```
7628      ;ML4AD
7629      ;
7630      ; ROUTINE DECLARATION SECTION
7631 027406 104414      TRAP 14
7632 027410 016700 166414  MOV ML.LUN,RO
7633 027414 104451      TRAP 51
7634 027416 104444      TRAP 44
7635 027420 062706 000006  ADD #6,SP
7636 027424 000207      2$: RTS PC
7637
7638      ; Routine Size: 84 words
7639      ; Maximum stack depth per invocation: 3 words
7644
7645
```

```
29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<
4514
4511
4466
```

7647 :ML4AD
7648 :
7649 :
7650 :
7651 :
7652 :
7653 :
7654 :
7655 :
7656 :
7657 :
7658 :
7659 :
7660 :
7661 :
7662 :
7663 :
7664 :
7665 :
7666 :
7667 :
7668 :
7669 :
7670 :
7671 :
7672 :
7673 :
7674 :
7675 :
7676 :
7677 :
7678 :
7679 :
7680 :
7681 :
7682 :
7683 :
7684 :
7685 :
7686 :
7687 :
7688 :
7689 :
7690 :
7691 :
7692 :
7693 :
7694 :
7695 :
7696 :
7697 :
7698 :

ROUTINE DECLARATION SECTION

routine WRT_TRANSFER (SIZE, DST, SRC) : novalue =
begin

!++
FUNCTIONAL DESCRIPTION
THIS ROUTINE WHEN CALLED WILL PERFORM
A WRITE TRANSFER TO THE REQUESTED
DESTINATION AND SOURCE ADDRESSES OF
LENGTH 'SIZE'.

FORMAL PARAMETERS:
SIZE: DETERMINES THE NUMBER OF WORDS
TO TRANSFER.
DST: DETERMINES THE DESTINATION ADRS
OF THE TRANSFER.
SRC: DETERMINES THE SOURCE ADRS OF THE
TRANSFER.

CLR_MBUS;
BAI = ONE;
MLWC = .SIZE;
MLDA = .DST;
MLBA = .SRC;
MLCS1 = write;

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (50)

if .SC
then
begin
ERRDF (148, INTER, DUMPER);
PRINTB (ONE_FMT, W_ERR);
DODU (.ML_LON);
DOCLN;
end;

!CLEAR THE MASS BUS BEFORE WE START
!FORCE THE TRANSFER TO STAY ON ONE BUS ADRS
!LOAD THE WORD COUNT SIZE
!LOAD THE DESTINATION ADRS
!LOAD THE SOURCE ADRS
!LOAD THE FUNCTION INTO CS1

do
0
until .DRY;

!DID LOADING THE FUNCTION CAUSE AN SC ERROR

if .SC
then
begin
ERRDF (148, INTER, DUMPER);
PRINTB (ONE_FMT, W_ERR);
DODU (.ML_LON);
DOCLN;
end;

!REPORT THE ERROR
!TELL WHAT THE ERROR IS
!DROP THE UNIT
!EXIT THE PROGRAM

!DO NOTHING
!UNTIL THE DRIVE IS READY

!DID SC ERROR SET DURING THE TRANSFER

end;

!REPORT THE ERORR
!TELL WHAT THE ERROR IS
!DROP THE UNIT
!EXIT THE PROGRAM

7759
7760
7761
7766
7767
7768 :

:ML4AD
:

ROUTINE DECLARATION SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

4568 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (52)

```

7770 :ML4AD
7771 :
7772 :      INITIALIZATION CODE SECTION
7773 :      4569 %sbttl 'INITIALIZATION CODE SECTION'
7774 :      4570 :
7775 :      4571 :
7776 :      4572 BGNINIT:
7777 :      4573 :      INITIALIZATION CODE IS EXECUTED AT THE BEGINNING OF EACH
7778 :      4574 :      PASS, WHEN POWER DOWN/POWER UP HAS OCCURRED, OR WHEN THE
7779 :      4575 :      OPERATOR HAS ISSUED A START, RESTART OR CONTINUE COMMAND.
7780 :      4576 :      DURING INITIALIZATION, THE 'GPHARD' MACRO IS USED TO GET
7781 :      4577 :      P-TABLE INFORMATION FOR THE LOGICAL UNIT UNDER TEST. THE
7782 :      4578 :      NUMBER OF UNITS AVAILABLE FOR TESTING IS CONTAINED IN A
7783 :      4579 :      HEADER LOCATION ('LSUNIT').
7784 :      4580 :
7785 :      4581 local
7786 :      4582   OFFSET:
7787 :      4583 :
7788 :      4584 external
7789 :      4585   LSUNIT:
7790 :      4586 :
7791 :      4587 if not READEF (EF_CONTINUE)           !SKIP INIT CODE IF CONTINUE
7792 :      4588 then
7793 :      4589   begin
7794 :      4590     !START GPHARDS AT LUN 0 AND LOAD 'ML_REG'
7795 :      4591     if READEF (EF_START)
7796 :      4592     then
7797 :      4593       begin
7798 :      4594         ML_LUN = -1;
7799 :      4595         DROP_CNT = ZERO;
7800 :      4596         !SEE IF THIS IS THE VERY FIRST PASS
7801 :      4597         !THIS IS CATEGORY 1 CODE
7802 :      4598         !CLEAR THE PASS COUNTER
7803 :      4599         do
7804 :      4600           begin
7805 :      4601             ML_LUN = .ML_LUN + 1;
7806 :      4602             !INCREMENT LOGICAL UNIT NUMBER
7807 :      4603             if .ML_LUN geq .LSUNIT then DOCLN;
7808 :      4604             !START OVER IF ALL UNITS HAVE BEEN TESTED
7809 :      4605             !REPEAT THE GPHARD UNTIL A 0 IS RETURNED
7810 :      4606             !GET BASE RH ADDRESS FOR THIS UNIT
7811 :      4607             !GET RH TYPE FOR THIS UNIT
7812 :      4608             !GET RH VECTOR FOR THIS UNIT
7813 :      4609             !INIT OFF SET COUNT
7814 :      4610             !LOAD THE REGISTER ADDRESS FOR THIS UNIT INTO ML_REG
7815 :      4611             !LOAD THE REGISTER ADDRESS FOR THIS UNIT INTO ML_REG
7816 :      4612             begin
7817 :      4613               ML_REG [.COUNT, REGISTER_ADD] = .RH_ADD + .OFFSET;
7818 :      4614               OFFSET = .OFFSET + 2;
7819 :      4615             end;
7820 :      4616           end
7821 :      4617         end
7822 :      4618       else
7823 :      4619         begin
7824 :      4620           !IS THIS A NEW PASS

```


29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (52)

7826 :ML4AD
7827 :
7828 :
7829 :
7830 :
7831 :
7832 :
7833 :
7834 :
7835 :
7836 :
7837 :
7838 :
7839 :
7840 :
7841 :
7842 :
7843 :
7844 :
7845 :
7846 :
7847 :
7848 :
7849 :
7850 :
7851 :
7852 :
7853 :
7854 :
7855 :
7856 :
7857 :
7858 :
7859 :
7860 :
7861 :
7862 :
7863 :
7864 :
7865 :
7866 :
7867 :
7868 :
7869 :
7870 :
7871 :
7872 :
7873 :
7874 :
7875 :
7876 :
7877 :
7878 :
7879 :
7880 :

4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672

INITIALIZATION CODE SECTION

```

if READEF (EF_RESTART) then DROP_CNT = -1;      !RESET THE PASS COUNTER
if READEF (EF_NEW)                               !IS THIS A NEW PASS
then
begin                                             !THIS IS A NEW PASS
DROP_CNT = .DROP_CNT + 1;                       !INCREMENT THE PASS COUNT
if (.ONEPAS) and (.DROP_CNT eq 1)              !HAVE WE DONE ONE PASS YET
then
begin                                             !DROP THE UNIT ON THE SECOND PASS
DODU (.ML_LUN);                                !DROP THE UNIT
DOCLN;                                         !JUMP TO THE CLEAN UP CODE
end
else                                             !THE FIRST PASS IS NOT COMPLETED YET
ML_LUN = -1;                                   !RESET THE LUN POINTER
end;
do
begin
ML_LUN = .ML_LUN + 1;                           !IF NOT GET NEXT LUN PTABLE
if .ML_LUN geq .LSUNIT then DOCLN;            !START OVER IF ALL UNITS ARE TESTED
end
until (GPHARD (.ML_LUN, PTBL_PTR)) neq 0;      !REPEAT UNTIL A 0 IS RETURNED
end;
PAR DIS = .((.PTBL_PTR) + 12);                 !GET PARITY DISABLE FLAG
ML_DUT = .((.PTBL_PTR) + 10);                 !GET DRIVE NUMBER
OP_NUM ARR = .((.PTBL_PTR) + 6) - 1;          !GET OPERATOR NUMBER OF ARRAYS
GOOD_BLK = ZEROES;                           !INIT GOOD BLOCK TO BLOCK ZERO
ARR_T6 = ZEROES;                              !INIT ARRAY 16 TO ZERO
LST_ARR = ZEROES;                             !INIT LAST ARRAY TO ZERO
LST_BLK = ZEROES;                             !INIT LAST BLOCK TO ZERO
:
: DEFINE ERROR PRINTING THRESHOLD LIMIT
:
if .ERRTHR then LIMIT = 10 else LIMIT = %o'077777';
:
: CALCULATE THE MOS RAM SIZE RUN TIME
: PARAMETERS. 'EITHER 16K OR 64K'
:
if .((.PTBL_PTR) + 8) IS_SET                   !CALCULATE ML11 16K MOS RAM PARAMETERS
then
begin
DRIVE_TYPE = %o'000110';                       !EXPECTED DRIVE TYPE VALUE
W_C_SIZE = %o'140000';                         !WORD COUNT SIZE FOR 16K WORD XFER
end

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (52)

7882 :ML4AD
7883 :
7884 :
7885 :
7886 :
7887 :
7888 :
7889 :
7890 :
7891 :
7892 :
7893 :
7894 :
7895 :
7896 :
7897 :
7898 :
7899 :
7900 :
7901 :
7902 :
7903 :
7904 :
7905 :
7906 :
7907 :
7908 :
7909 :
7910 :
7911 :
7912 :
7913 :
7914 :
7915 :
7919 :
7920 :
7921 :
7922 :
7923 :
7927 :
7928 :
7929 :
7930 :
7931 :
7932 :
7933 :
7934 :
7935 :
7936 :

INITIALIZATION CODE SECTION

```

4673 RAS_INC = %0'200';
4674 CHIP_SIZ = 16;
4675 ARR_INC = %0'1000';
4676 ARR_16<9, 4> = %0'17';
4677 LST_ARR<9, 4> = .OP_NUM_ARR;
4678 LST_BLK<9, 4> = .OP_NUM_ARR;
4679 LST_BLK = .LST_BLK or %0'777';
4680 end
4681 else
4682 begin
4683 DRIVE_TYPE = %0'000111';
4684 W_C_SIZE = %0'000000';
4685 RAS_INC = %0'1000';
4686 CHIP_SIZ = 64;
4687 ARR_INC = %0'4000';
4688
4689 : VERSION CZMLAD CHANGED %0'37' TO %0'17'
4690 :
4691 ARR_16<11, 4> = %0'17';
4692 LST_ARR<11, 4> = .OP_NUM_ARR;
4693 LST_BLK<11, 4> = .OP_NUM_ARR;
4694 LST_BLK = .LST_BLK or %0'3777';
4695 end;
4696
4697 if ((.PTBL_PTR) + 2) eql %0'70' then LST_DUT_REG = 21 else LST_DUT_REG = 19;
4698
4699 PRINTB (FMT_17, .ML_LUN);
4700 CLR_MBUS;
4701 end;
4702
4703 ENDINIT;

```

```

!RAS INCREMENT FOR 16K RAMS
!CHIP SIZE
!ARRAY INCREMENT
!ARRAY 16
!LAST ARRAY
!LAST BLOCK

!CALCULATE ML11 64K MOS RAM PARAMETERS

!EXPECTED DRIVE TYPE VALUE
!WORD COUNT SIZE FOR 64K WORD XFER
!RAS INCREMENT FOR 64K RAMS
!CHIP SIZE
!ARRAY INCREMENT

!ARRAY 16
!LAST ARRAY
!LAST BLOCK

!TELL OPERATOR WHICH UNIT IS BEING TESTED
!CLEAR MASS BUS

```

.GLOBL L\$UNIT

```

.LIMIT: .SBTTL LINIT INITIALIZATION CODE SECTION
JSR R1,$SAVE3
MOV #36,R0
TRAP 47
BHS 1$
RTS PC
1$: MOV #40,R0
TRAP 47
BHS 5$
MOV #-1,ML_LUN
CLR DROP.CNT

```

4567
4587
4591
4594
4595

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

```

8050      :ML4AD
8051      :
8052      :
8053 030524 006303      ASL      R3
8054 030526 042703 103777 BIC      #103777,R3
8055 030532 042767 074000 163260 BIC      #74000,LST.ARR
8056 030540 050367 163254      BIS      R3,LST.ARR
8057 030544 016703 163232      MOV      OP.NUM.ARR,R3
8058 030550 000303      SWAB     R3
8059 030552 006303      ASL      R3
8060 030554 006303      ASL      R3
8061 030556 006303      ASL      R3
8062 030560 042703 103777 BIC      #103777,R3
8063 030564 042767 074000 163222 BIC      #74000,LST.BLK
8064 030572 050367 163216      BIS      R3,LST.BLK
8065 030576 052767 003777 163210 BIS      #3777,LST.BLK
8066 030604 016701 163170 14$: MOV      PTBL.PTR,R1
8067 030610 026127 000002 000070 CMP      2(R1),#70
8068 030616 001004      BNE      15$
8069 030620 012767 000025 164530 MOV      #25,LST.DUT.REG
8070 030626 004403      BR       16$
8071 030630 012767 000023 164520 15$: MOV      #23,LST.DUT.REG
8072 030636 016746 165166 16$: MOV      ML.LUN,-(SP)
8073 030642 012746 007526      MOV      #FMT.17,-(SP)
8074 030646 012746 000002      MOV      #2,-(SP)
8075 030652 010600      MOV      SP,R0
8076 030654 104414      TRAP     14
8077 030656 152777 000040 164560 BISB     #40,2ML.REG+40
8078 030664 016703 165142      MOV      ML.DUT,R3
8079 030670 042703 177770      BIC      #177770,R3
8080 030674 142777 000007 164542 BICB     #7,2ML.REG+40
8081 030702 150377 164536      BISB     R3,2ML.REG+40
8082 030706 062706 000006      ADD      #6,SP
8083 030712 000207      RTS      PC
8084
8085      : Routine Size: 274 words
8086      : Maximum stack depth per invocation: 7 words
8091
8092
8096
8097
8101 030714 004767 176730      .SBTTL  L$INIT INITIALIZATION CODE SECTION
8102 030720 104411      L$INIT::JSR PC,LINIT
8103 030722 000207      TRAP     11
8104      RTS      PC
  
```

4693

4694

4697

4699

4589

4567

4701

8106 :ML4AD
8107 :
8108 : INITIALIZATION CODE SECTION
8109 : Routine Size: 4 words
8110 : Maximum stack depth per invocation: 0 words
8115
8116
8117 : 4704 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (53)

```
8119 :ML4AD
8120 :
8121 :
8122 : 4705 %SBTTL 'TEST CODE SECTION'
8123 : 4706
8124 : 4707 !
8125 : 4708 BGNTST;
8126 : 4709
8127 : 4710 !++
8128 : 4711 TEST NUMBER: TST 1
8129 : 4712
8130 : 4713 TEST NAME: MASS BUS READY TEST
8131 : 4714
8132 : 4715 TEST DESCRIPTION:
8133 : 4716 TEST THE RH CONTROLLER FOR EXISTANCE
8134 : 4717 BY:
8135 : 4718
8136 : 4719 1. WRITTING ONES TO THE RH CS2 REGISTER
8137 : 4720 RESULTING IN A MASS BUS CLEAR
8138 : 4721
8139 : 4722 2. THEN READ THE CS2 REGISTER FOR CLEAR DATA
8140 : 4723
8141 : 4724 !--
8142 : 4725
8143 : 4726 local
8144 : 4727 DODU_FLG; !DROP UNIT FLAG
8145 : 4728
8146 : 4729 DODU_FLG = ZERO; !CLEAR THE DROP UNIT FLAG
8147 : 4730 BGNSUB;
8148 : 4731 CLR MBUS;
8149 : 4732 MLCS1 = ZEROES; !CLEAR OUT CS1
8150 : 4733 MLCS2 = %o'177770'; !LOAD CS2 WITH ONES AND FORCE A MBUS CLEAR
8151 : 4734 DELAY (ONE_US);
8152 : 4735
8153 : 4736 if ((.MLCS2) and (%o'177670')) neq ZERO !SEE IF CS2 GOT CLEARED
8154 : 4737 then
8155 : 4738 begin !REPORT ERROR IF NOT CLEARED
8156 : 4739 ERRDF (119, RH_ERROR, DUMPER);
8157 : 4740 PRINTB (THR_FMT, REG_17, FNC_23, PHR_4);
8158 : 4741 DODU_FLG = ONE;
8159 : 4742 end;
8160 : 4743
8161 : 4744 ENDSUB;
8162 : 4745
8163 : 4746 if .DODU_FLG IS_SET !DROP THE UNIT IF DODU_FLG IS SET
8164 : 4747 then
8165 : 4748 begin
8166 : 4749 DODU (.ML_LUN);
8167 : 4750 DOCLN;
8168 : 4751 end;
8169 : 4752
8170 : 4753 ENDTST;
```

Address	OpCode	Op1	Op2	Op3	Op4	Label	Comment	Address
8175								
8176								
8177								
8178								
8179								
8183	030724	004167	153610			\$T1:	.SBTTL \$T1 TEST CODE SECTION	
8184	030730	005746					JSR R1,\$SAVE2	4703
8185	030732	005002					TST -(SP)	
8186	030734	104402				1\$:	CLR R2	: DODU.FLG 4729
8187	030736	152777	000040	164500			TRAP 2	
8188	030744	016701	165062				BISB #40,@ML.REG+40	4730
8189	030750	042701	177770				MOV ML,DUT,R1	
8190	030754	142777	000007	164462			BIC #177770,R1	
8191	030762	150177	164456				BICB #7,@ML.REG+40	
8192	030766	005077	164412				BISB R1,@ML.REG+40	
8193	030772	012777	177770	164444			CLR @ML.REG	: 4732
8194	031000	012700	000001				MOV #-10,@ML.REG+40	: 4733
8195	031004	001410				2\$:	MOV #1,R0	: *,\$STMP2 4734
8196	031006	016701	151104				BEQ 5\$	
8197	031012	001403					MOV LSDLY,R1	: *,\$STMP1
8198	031014	005016				3\$:	BEQ 4\$	
8199	031016	005301					CLR (SP)	: \$STMP
8200	031020	001375					DEC R1	: \$STMP1
8201	031022	005300				4\$:	BNE 3\$	
8202	031024	000767					DEC R0	: \$STMP2
8203	031026	032777	177670	164410		5\$:	BR 2\$	
8204	031034	001424					BIT #-110,@ML.REG+40	: 4736
8205	031036	104455					BEQ 6\$	4739
8206	031040	000167					TRAP 5\$	
8207	031042	013302					.WORD 167	
8208	031044	026302					.WORD RH.ERROR	
8209	031046	012746	011676				.WORD DUMPER	
8210	031052	012746	012464				MOV #PHR.4,-(SP)	: 4740
8211	031056	012746	012642				MOV #FNC.2\$,-(SP)	
8212	031062	012746	010256				MOV #REG.17,-(SP)	
8213	031066	012746	000004				MOV #THR.FMT,-(SP)	
8214	031072	010600					MOV #4,-(SP)	
8215	031074	104414					MOV SP,R0	: SP,*
8216	031076	012702	000001				TRAP 14	
8217	031102	062706	000012				MOV #1,R2	: *,DODU.FLG 4741
8218	031106	104467				6\$:	ADD #12,SP	: 4738
8219	031110	006000					TRAP 67	: 4742
8220	031112	103710					ROR R0	
8221	031114	005302					BLO 1\$	
8222	031116	001004					DEC R2	: DODU.FLG 4746
8223	031120	016700	164704				BNE 7\$	
8224	031124	104451					MOV ML.LUN,R0	: 4749
8225	031126	104444					TRAP 51	
8226	031130	005726				7\$:	TRAP 44	
8227	031132	000207					TST (SP)+	: 4703
8228							RTS PC	
8229								

: Routine Size: 68 words

8231
8232
8233
8234
8239
8240
8244
8245
8249 031134
8250 031134 004767 177564
8251 031140 104466
8252 031142 006000
8253 031144 103773
8254 031146 000207
8255
8256
8257
8262
8263
8264 :

:ML4AD
:
TEST CODE SECTION
: Maximum stack depth per invocation: 9 words

.SBTTL T1: TEST CODE SECTION
T1::
1\$: JSR PC,\$T1
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

4754 !<BLF/PAGE>

4751

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (54)

```

8266 :ML4AD
8267 :
8268 :
8269 : 4755 !
8270 : 4756 BGNTST:
8271 : 4757
8272 : 4758 !++
8273 : 4759 ! TEST NUMBER: TST 2
8274 : 4760 !
8275 : 4761 ! TEST NAME: MASS BUS HANDSHAKE TEST
8276 : 4762 !
8277 : 4763 ! TEST DESCRIPTION:
8278 : 4764 ! TEST MASS BUS TO UNIBUS COMMUNICATIONS
8279 : 4765 ! VIA THE CONTROL BUS BY:
8280 : 4766 !
8281 : 4767 ! 1. READING RH AND DRIVE REGISTERS AND
8282 : 4768 ! TEST THE NON EXISTANT DRIVE 'NED'
8283 : 4769 ! BIT
8284 : 4770 !
8285 : 4771 !--
8286 : 4772
8287 : 4773 local
8288 : 4774 SAVE,
8289 : 4775 DODU_FLG;
8290 : 4776
8291 : 4777 CLR THRESHOLD;
8292 : 4778 DODU_FLG = ZERO;
8293 : 4779
8294 : 4780 incr REG_SEL from 0 to .LST_DUT_REG do
8295 : 4781 begin
8296 : 4782 BGNSUB;
8297 : 4783 CLR MBUS;
8298 : 4784 SAVE = .ML_REG [.REG_SEL, REGISTER_ADD];
8299 : 4785 DELAY (ONE_DS);
8300 : 4786
8301 : 4787 if .NED IS_SET
8302 : 4788 then
8303 : 4789 begin
8304 : 4790 CMP THRESHOLD;
8305 : 4791 ERRDF (120, RH_ERROR, DUMPER);
8306 : 4792 PRINTB (FIV_FMT, WRD 62, PHR 5, WRD 12, WRD 52, FNC 6);
8307 : 4793 PRINTB (FMT 11, .ML_REG [.REG_SEL, REGISTER_ADD]);
8308 : 4794 DODU_FLG = ONE;
8309 : 4795 end;
8310 : 4796
8311 : 4797 ENDSUB;
8312 : 4798 end;
8313 : 4799
8314 : 4800 if .DODU_FLG IS_SET
8315 : 4801 then
8316 : 4802 begin
8317 : 4803 DODU (.ML_LUN);
8318 : 4804 DOCLN;
8319 : 4805 end;
8320 : 4806

```

```

!TEMPORARY SAVE LOCATION
!DROP UNIT FLAG
!CLEAR ERROR PRINT THRESHOLD
!TEST ALL PRESENT RH REGISTERS
!START OF SCOPE LOOP
!READ THE REGISTER
!DELAY ONE MICRO SECOND
!DID READ CAUSE THE NED BIT TO SET
!REPORT AN ERROR IF SET
!COMPARE ERROR PRINT THRESHOLD
!END OF SCOPE LOOP

```


8378
 8379
 8380
 8381 031362 012746 000006
 8382 031366 010600
 8383 031370 104414
 8384 031372 016316 015404
 8385 031376 012746 007170
 8386 031402 012746 000002
 8387 031406 010600
 8388 031410 104414
 8389 031412 012701 000001
 8390 031416 062706 000022
 8391 031422 104467
 8392 031424 006000
 8393 031426 103667
 8394 031430 005202
 8395 031432 020216
 8396 031434 003657
 8397 031436 005301
 8398 031440 001004
 8399 031442 016700 164362
 8400 031446 104451
 8401 031450 104444
 8402 031452 062706 000006
 8403 031456 000207

:ML4AD

TEST CODE SECTION

```

MOV #6,-(SP)
MOV SP,R0           ; SP,*
TRAP 14
MOV ML.REG(R3),(SP)
MOV #FMT.11,--(SP)
MOV #2,-(SP)
MOV SP,R0           ; SP,*
TRAP 14
MOV #1,R1           ; *,DODU.FLG
ADD #22,SP
7$: TRAP 67
ROR R0
BLO 2$
8$: INC R2
9$: CMP R2,(SP)     ; REG.SEL
BLE 1$             ; REG.SEL,*
DEC R1             ; DODU.FLG
BNE 10$
MOV ML.LUN,R0
TRAP 51
TRAP 44
10$: ADD #6,SP
RTS PC
  
```

4793

4794

4789

4795

4780

4800

4803

4753

: Routine Size: 100 words
 : Maximum stack depth per invocation: 18 words

.SBTTL T2 TEST CODE SECTION

T2::
 1\$:

```

JSR PC,ST2
TRAP 66
ROR R0
BLO 1$
RTS PC
  
```

4805

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

8404
 8405
 8406
 8411
 8412
 8416
 8417
 8421 031460
 8422 031460 004767 177464
 8423 031464 104466
 8424 031466 006000
 8425 031470 103773
 8426 031472 000207
 8427
 8428
 8429

8438
8439
8440 : 4808 !<BLF/PAGE>

8442 :ML4AD
8443 :
8444 :
8445 :
8446 :
8447 :
8448 :
8449 :
8450 :
8451 :
8452 :
8453 :
8454 :
8455 :
8456 :
8457 :
8458 :
8459 :
8460 :
8461 :
8462 :
8463 :
8464 :
8465 :
8466 :
8467 :
8468 :
8469 :
8470 :
8471 :
8472 :
8473 :
8474 :
8475 :
8476 :
8477 :
8478 :
8479 :
8480 :
8481 :
8482 :
8483 :
8484 :
8485 :
8486 :
8487 :
8488 :
8489 :
8490 :
8491 :
8495 :
8496 :

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 3

TEST NAME: DRIVE PRESENT TEST

TEST DESCRIPTION:

THIS TEST READS THE DESIRED SECTOR
ADDRESS REGISTER OF THE DRIVE UNDER
TEST, DELAYS 100 US, THEN
READS THE NED BIT OF MLCS2

IF SET, AN ERROR MESSAGE IS
PRINTED AND THE UNIT IS DROPPED

--

local

DODU_FLG,
SAVE;

DODU_FLG = ZERO;

BGNSUB;

CLR MBUS;

SAVE = .MLDA;

DELAY (ONE_US);

if .NED IS_SET

then

begin

ERRDF (1, ASYNC, DUMPER);

PRINTB (ONE_FMT, PHR_3);

DODU_FLG = ONE;

end;

ENDSUB;

if .DODU_FLG IS_SET

then

begin

DODU (.ML_LUN);

DOCLN;

end;

ENDTST;

29-Mar-1982 16:23:04

29-Mar-1982 16:21:03

TCOS-20 Bliss-16 V2(212)

PA:<NEALE>ML4AD.BLI.4 (55)

!

!DROP UNIT FLAG
!TEMP STORAGE LOCATION

!READ A DRIVE REGISTER
!DELAY 1 US

!TEST THE NED BIT

!IF SET THEN REPORT ERROR AND SET DODU_FLG

!DROP THIS UNIT IF DODU IS_SET

.SBTTL \$T3 TEST CODE SECTION

Address	Label	Instruction	Comment	Time	Page
8498				29-Mar-1982 16:23:04	TOPS
8499	:ML4AD		TEST CODE SECTION	29-Mar-1982 16:21:03	PA:<
8500					
8504	ST3:	JSR R1,\$SAVE4			4807
8505		TST -(SP)			
8506		CLR R3	: DODU.FLG		4832
8507	1\$:	TRAP 2			
8508		BISB #40,@ML.REG+40			4833
8509		MOV ML.DUT,R2			
8510		BIC #177770,R2			
8511		BICB #7,@ML.REG+40			
8512		BISB R2,@ML.REG+40			
8513		MOV @ML.REG+30,R4	: *.SAVE		4835
8514		MOV #1,R1	: *.SSTMP2		4836
8515	2\$:	BEQ 5\$			
8516		MOV LSDLY,R2	: *.SSTMP1		
8517		BEQ 4\$			
8518	3\$:	CLR (SP)	: SSTMP		
8519		DEC R2	: SSTMP1		
8520		BNE 3\$			
8521	4\$:	DEC R1	: SSTMP2		
8522		BR 2\$			
8523	5\$:	BIT #10000,@ML.REG+40			4838
8524		BEQ 6\$			
8525		TRAP 55			4841
8526		.WORD 1			
8527		.WORD ASYNC			
8528		.WORD DUMPER			
8529		MOV #PHR.3,-(SP)			4842
8530		MOV #ONE.FMT,-(SP)			
8531		MOV #2,-(SP)			
8532		MOV SP,R0	: SP,*		
8533		TRAP 14			
8534		MOV #1,R3	: *.DODU.FLG		4843
8535		ADD #6,SP	:		4840
8536	6\$:	TRAP 67	:		4844
8537		ROR R0			
8538		BLO 1\$			
8539		DEC R3	: DODU.FLG		4848
8540		BNE 7\$			
8541		MOV ML.LUN,R0	:		4851
8542		TRAP 51			
8543		TRAP 44			
8544	7\$:	TST (SP)+	:		4807
8545		RTS PC			
8546					
8547					
8548					

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

8554 :ML4AD
8555 :
8556 :
8560 :
8561 :
8565 031666
8566 031666 004767 177602
8567 031672 104466
8568 031674 006000
8569 031676 103773
8570 031700 000207
8571 :
8572 :
8573 :
8578 :
8579 :
8580 :

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 BLI.S-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (55)

.SBTTL T3 TEST CODE SECTION

T3::
1\$:

JSR PC,\$T3
TRAP 66
ROR R0
BLO 1\$
RTS PC

4853

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

4856 !<BLF/PAGE>

8582 :ML4AD

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (56)

TEST CODE SECTION

```

8583 :
8584 :
8585 : 4857 !
8586 : 4858 BGNTST;
8587 : 4859
8588 : 4860 !++
8589 : 4861 TEST NUMBER: TST 4
8590 : 4862
8591 : 4863 TEST NAME: DRIVE SELECTION TEST
8592 : 4864
8593 : 4865 TEST DESCRIPTION:
8594 : 4866 THIS TEST TESTS FOR UNIQUE DRIVE SELECTION BY WRITING THE DRIVE
8595 : 4867 UNDER TEST (DUT) DRIVE NUMBER INTO ITS DSA REG. THEN WRITING THE DRIVE
8596 : 4868 NUMBERS OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REGISTERS
8597 : 4869 AND READING THE DUT DSA FOR ITS DRIVE NUMBER.
8598 : 4870
8599 : 4871 IMPLICIT INPUTS:
8600 : 4872 ML_DUT
8601 : 4873 LOADED DURING THE INITIALIZATION CODE AND CONTAINS THE DRIVE
8602 : 4874 NUMBER OF THE DRIVE PRESENTLY BEING TESTED.
8603 : 4875 !--
8604 : 4876
8605 : 4877 local
8606 : 4878 SAVE; !TEMPORARY SAVE LOCATION
8607 : 4879
8608 : 4880 BGNSUB;
8609 : 4881 CLR MBUS;
8610 : 4882 MLDA = .ML_DUT; !LOAD THIS DRIVES DRIVE NO. INTO ITS DSA REG
8611 : 4883 SAVE = .MLDA; !READ THE REGISTER BACK
8612 : 4884
8613 : 4885 if .SAVE neq .ML_DUT !SEE IF DSA HAS DRIVE NUMBER
8614 : 4886 then
8615 : 4887 begin
8616 : 4888 ERRDF (2, INTER, DUMPER); !ERROR AND EXIT TEST IF DSA NEQ DRIVE NUM
8617 : 4889 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 3, WRD 37, WRD 13, REG_6);
8618 : 4890 PRINTB (FMT_2, .ML_DUT, .SAVE, (.ML_DUT xof .SAVE));
8619 : 4891 EXIT_TST;
8620 : 4892 end;
8621 : 4893
8622 : 4894 incr DRV_SEL from 0 to 7 do !WRITE DRV NO OF OTHER DRIVES INTO THEIR RESPECTIVE DSA REG.
8623 : 4895
8624 : 4896 if .DRV_SEL neq .ML_DUT !SKIP IF .DRV_SEL EQL TO THE DRIVE UNDER TEST (DUT)
8625 : 4897 then
8626 : 4898 begin
8627 : 4899 DRV_NUM = .DRV_SEL; !SELECT DRIVE TO BE WRITTEN TO
8628 : 4900 MLDA = .DRV_SEL; !WRITE DRIVE SEL NO. INTO ITS DSA REG
8629 : 4901 DELAY (ONE_OS); !DELAY 1 US
8630 : 4902 end;
8631 : 4903
8632 : 4904 DRV_NUM = .ML_DUT; !SELECT THE DUT
8633 : 4905 SAVE = .MLDA; !READ ITS DSA REG
8634 : 4906
8635 : 4907 if .SAVE neq .ML_DUT !SEE IF WRITTING TO OTHER DRIVES CHANGE ITS VALUE
8636 : 4908 then

```

8638 :ML4AD
8639 :
8640 :
8641 :
8642 :
8643 :
8644 :
8645 :
8646 :
8647 :
8648 :
8652 :
8653 :

TEST CODE SECTION

```
begin
ERRDF (3, ASYNC, DUMPER);
PRINTB (THR_FMT, FNC 3, WRD 37, WRD_14);
PRINTB (FMT_1, .ML_DOT, .SAVE);
end;
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (56)

!ERROR AND SET DODU_FLG IF CHANGED

ENDSUB:
ENDTST:

Address	Hex	Hex	Hex	Label	Instruction	Comment	Address
8657	031702	004167	152646	ST4:	.SBTTL	ST4 TEST CODE SECTION	
8658	031706	005746			JSR	R1,\$SAVE3	
8659	031710	104402			TST	-(SP)	4855
8660	031712	152777	000040	1S:	TRAP	2	
8661	031720	016702	164106		BISB	#40,@ML.REG+40	4878
8662	031724	042702	177770		MOV	ML.DUT,R2	4880
8663	031730	142777	000007	163524	BIC	#177770,R2	
8664	031736	150277	163502		BICB	#7,@ML.REG+40	
8665	031742	016777	164064	163464	BISB	R2,@ML.REG+40	
8666	031750	017703	163460		MOV	ML.DUT,@ML.REG+30	
8667	031754	020367	164052		MOV	@ML.REG+30,R3	4882
8668	031760	001451			CMP	R3,ML.DUT	4883
8669	031762	104455			BEQ	2\$	4885
8670	031764	000002			TRAP	55	
8671	031766	013114			.WORD	2	4888
8672	031770	026302			.WORD	INTER	
8673	031772	012746	012540		.WORD	DUMPER	
8674	031776	012746	010660		MOV	#REG.6,-(SP)	
8675	032002	012746	011122		MOV	#WRD.13,-(SP)	4889
8676	032006	012746	012172		MOV	#WRD.37,-(SP)	
8677	032012	012746	010650		MOV	#FNC.3,-(SP)	
8678	032016	012746	011676		MOV	#WRD.12,-(SP)	
8679	032022	012746	010322		MOV	#PHR.4,-(SP)	
8680	032026	012746	000007		MOV	#SIX.FMT,-(SP)	
8681	032032	010600			MOV	#7,-(SP)	
8682	032034	104414			MOV	SP,R0	: SP,*
8683	032036	016716	163770		TRAP	14	
8684	032042	010302			MOV	ML.DUT,(SP)	
8685	032044	041602			MOV	R3,R2	: SAVE,*
8686	032046	040316			BIC	(SP),R2	
8687	032050	050216			BIC	R3,(SP)	: SAVE,*
8688	032052	010346			BIS	R2,(SP)	
8689	032054	016746	163752		MOV	R3,-(SP)	: SAVE,*
8690	032060	012746	006506		MOV	ML.DUT,-(SP)	
8691	032064	012746	000004		MOV	#FMT.2,-(SP)	
8692	032070	010600			MOV	#4,-(SP)	
					MOV	SP,R0	: SP,*

Address	OpCode	Operand1	Operand2	Label	Instruction	Comments	Line No.
8694							
8695							
8696							
8697	032072	104414			TRAP 14		
8698	032074	104463			TRAP 63		
8699	032076	062706	000030		ADD #30,SP		
8700	032102	000513			BR 9\$		4885
8701	032104	005002		2\$:	CLR R2	DRV.SEL	4887
8702	032106	020267	163720	3\$:	CMP R2,ML.DUT	DRV.SEL,*	4894
8703	032112	001425			BEQ 7\$		4896
8704	032114	010201			MOV R2,R1	DRV.SEL,*	
8705	032116	042701	177770		BIC #177770,R1		4899
8706	032122	142777	000007	163314	BICB #7,@ML.REG+40		
8707	032130	150177	163310		BISB R1,@ML.REG+40		
8708	032134	010277	163274		MOV R2,@ML.REG+30		
8709	032140	012700	000001		MOV #1,R0	DRV.SEL,*	4900
8710	032144	001410		4\$:	BEQ 7\$	*,\$:TMP2	4901
8711	032146	016701	147744		MOV LSDLY,R1	*,\$:TMP1	
8712	032152	001403			BEQ 6\$		
8713	032154	005016		5\$:	CLR (SP)	\$:TMP	
8714	032156	005301			DEC R1	\$:TMP1	
8715	032160	001375			BNE 5\$		
8716	032162	005300		6\$:	DEC R0	\$:TMP2	
8717	032164	000767			BR 4\$		
8718	032166	005202		7\$:	INC R2	DRV.SEL	
8719	032170	020227	000007		CMP R2,#7	DRV.SEL,*	4894
8720	032174	003744			BLE 3\$		
8721	032176	016702	163630		MOV ML.DUT,R2		4904
8722	032202	042702	177770		BIC #177770,R2		
8723	032206	142777	000007	163230	BICB #7,@ML.REG+40		
8724	032214	150277	163224		BISB R2,@ML.REG+40		
8725	032220	017703	163210		MOV @ML.REG+30,R3	*,\$:SAVE	4905
8726	032224	020367	163602		CMP R3,ML.DUT	SAVE,*	4907
8727	032230	001433			BEQ 8\$		
8728	032232	104455			TRAP 55		
8729	032234	000003			.WORD 3		4910
8730	032236	012706			.WORD ASYNC		
8731	032240	026302			.WORD DUMPER		
8732	032242	012746	010664		MOV #WRD.14,-(SP)		
8733	032246	012746	011122		MOV #WRD.37,-(SP)		4911
8734	032252	012746	012172		MOV #FNC.3,-(SP)		
8735	032256	012746	010256		MOV #THR.FMT,-(SP)		
8736	032262	012746	000004		MOV #4,-(SP)		
8737	032266	010600			MOV SP,R0	SP,*	
8738	032270	104414			TRAP 14		
8739	032272	010316			MOV R3,(SP)	SAVE,*	
8740	032274	016746	163532		MOV ML.DUT,-(SP)		4912
8741	032300	012746	006442		MOV #FMT.1,-(SP)		
8742	032304	012746	000003		MOV #3,-(SP)		
8743	032310	010600			MOV SP,R0	SP,*	
8744	032312	104414			TRAP 14		
8745	032314	062706	000020		ADD #20,SP		
8746	032320	104467		8\$:	TRAP 67		4909
8747	032322	006000			ROR R0		4913
8748	032324	103002			BHIS 9\$		

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

8750
8751
8752
8753 032326 000167 177356
8754 032332 005726
8755 032334 000207
8756
8757
8758
8763
8764
8768
8769

:ML4AD
:
TEST CODE SECTION

9\$: JMP 1\$
TST (SP)+
RTS PC ;

4855

: Routine Size: 142 words
: Maximum stack depth per invocation: 17 words

8773 032336
8774 032336 004767 177340
8775 032342 104466
8776 032344 006000
8777 032346 103773
8778 032350 000207
8779
8780
8781
8786
8787
8788 :

.SBTTL T4 TEST CODE SECTION

T4::
1\$: JSR PC,ST4
TRAP 66 ;
ROR R0
BLO 1\$
RTS PC

4915

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

4917 !<BLF/PAGE>

8790 :ML4AD

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (57)

8791 :
8792 :
8793 : 4918
8794 : 4919
8795 : 4920
8796 : 4921
8797 : 4922
8798 : 4923
8799 : 4924
8800 : 4925
8801 : 4926
8802 : 4927
8803 : 4928
8804 : 4929
8805 : 4930
8806 : 4931
8807 : 4932
8808 : 4933
8809 : 4934
8810 : 4935
8811 : 4936
8812 : 4937
8813 : 4938
8814 : 4939
8815 : 4940
8816 : 4941
8817 : 4942
8818 : 4943
8819 : 4944
8820 : 4945
8821 : 4946
8822 : 4947
8823 : 4948
8824 : 4949
8825 : 4950
8826 : 4951
8827 : 4952
8828 : 4953
8829 : 4954
8830 : 4955
8831 : 4956
8832 : 4957
8833 : 4958
8834 : 4959
8835 : 4960
8836 : 4961
8837 : 4962
8838 : 4963
8839 : 4964
8840 : 4965
8841 : 4966
8842 : 4967
8843 : 4968
8844 : 4969

! BGNTST;

!++

TEST NUMBER: TST 5

TEST NAME: READ WRITE REG ONES/ZEROES TEST

TEST DESCRIPTION:

THIS TEST WRITES AND READS A DATA PATTERN OF ALL ONES AND ZEROES TO ALL OF THE ML11'S READ / WRITE REGISTERS.

ROUTINES WRT_REG AND RD_REG ACCEPT ARGUMENTS TO FURTHER SELECT ROUTINES WHICH ACTUALLY PERFORMS THE READING AND WRITING OF THE REGISTERS.

THE UNIT IS DROPPED ON DETECTED ERRORS.

IMPLICIT INPUTS:

WT_DATA

LOADED BY READ REGISTER ROUTINES AND CONTAINS THE DATA PATTERN WRITTEN TO THE REGISTERS (REPRESENTS GOOD DATA).

RD_DATA

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

Local

ERR_FLG,

TST_PAT,

index,

DODU_FLG;

!ERROR FLAG PASSED TO ROUTINES

!TEST PATTERN

!POINTS TO REGISTER PRESENTLY BEING TESTED

!DROP UNIT FLAG

CLR_THRESHOLD:

DODU_FLG = ZERO;

TST_PAT = ONES;

!CLEAR ERROR PRINT THRESHOLD

!LOAD TEST PAT WITH ONES

incr TWICE from 0 to 1 do

begin

!REPEAT LOOP TWICE

incr REG_SEL from 0 to 7 do

begin

!TEST ELEVEN WRITE/READ REGISTERS

BGN SUB;

CLR_MBUS;

WRT_REG (.TST_PAT, .REG_SEL, index);

RD_REG (.TST_PAT, .REG_SEL, ERR_FLG);

!WRITE TO THE REGISTER

!READ THE REGISTER

if .ERR_FLG IS_SET

then

!SEE IF READ FOUND AN ERROR

begin

CMP_THRESHOLD;

!IF ERROR FLAG IS SET THEN ERROR AND SET DODU_FLG

!COMPARE ERROR PRINT THRESHOLD

```

8846 :ML4AD
8847 :
8848 :
8849 : 4970
8850 : 4971
8851 : 4972
8852 : 4973
8853 : 4974
8854 : 4975
8855 : 4976
8856 : 4977
8857 : 4978
8858 : 4979
8859 : 4980
8860 : 4981
8861 : 4982
8862 : 4983
8863 : 4984
8864 : 4985
8865 : 4986
8866 : 4987
8867 : 4988
8868 : 4989
8869 : 4990
8870 : 4991
8871 : 4992
8872 : 4993
8873 : 4994
8874 : 4995
8875 : 4996
8876 : 4997
8877 : 4998
8878 : 4999
8879 : 5000
8880 : 5001
8884 :
8885 :
8889 032352 004167 152214
8890 032356 024646
8891 032360 005067 163012
8892 032364 005004
8893 032366 012702 177777
8894 032372 005001
8895 032374 005003
8896 032376 104402
8897 032400 152777 000040 163036
8898 032406 016700 163420
8899 032412 042700 177770
8900 032416 142777 000007 163020

TEST CODE SECTION
selectone .REG_SEL of
set
[0 to 3] :
ERRDF (4, ASYNC, DUMPER);
[4 to 5] :
ERRDF (4, SYNC, DUMPER);
[6 to 7] :
ERRDF (4, ARR_DAT, DUMPER);
tes:
PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
PRINTB (FMT 16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
DODU_FLG = ONE;
end;
ENDSUB;
end;
TST_PAT = not .TST_PAT;
end;
if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;
ENDTST;

!SELECT WHICH MODULE FAILED
!ASYNC MODULE FAILURE
!SYNC MODULE FAILURE
!ARRAY DATA MODULE FAILURE
!REPEAT AGAIN WITH COMPLIMENT DATA
!DROP THIS UNIT IF THE DODU_FLG IS_SET

.SBTTL $T5 TEST CODE SECTION
$T5: JSR R1,$SAVE4
CMP -(SP),-(SP)
CLR P.CNT
CLR R4
MOV #-1,R2
CLR R1
1$: CLR R3
2$: TRAP 2
BISB #40,@ML.REG+40
MOV ML.DUT,R0
BIC #177770,R0
BICB #7,@ML.REG+40

DODU.FLG
*.TST.PAT
TWICE
REG.SEL

4916
4949
4952
4953
4955
4958
4959
4960

```


9014
9015
9016
9017 033012 104466
9018 033014 006000
9019 033016 103773
9020 033020 000207

:ML4AD
:
TEST CODE SECTION

TRAP 66
ROR R0
BLO 1\$
RTS PC

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

9021
9022
9023
9028
9029
9030 : 5002 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (58)

9032 :ML4AD
9033 :
9034 :
9035 :
9036 :
9037 :
9038 :
9039 :
9040 :
9041 :
9042 :
9043 :
9044 :
9045 :
9046 :
9047 :
9048 :
9049 :
9050 :
9051 :
9052 :
9053 :
9054 :
9055 :
9056 :
9057 :
9058 :
9059 :
9060 :
9061 :
9062 :
9063 :
9064 :
9065 :
9066 :
9067 :
9068 :
9069 :
9070 :
9071 :
9072 :
9073 :
9074 :
9075 :
9076 :
9077 :
9078 :
9079 :
9080 :
9081 :
9082 :
9083 :
9084 :
9085 :
9086 :

TEST CODE SECTION

!
BGNTST;
!++
TEST NUMBER: TST 6
TEST NAME: READ WRITE REG SHIFTING ONES/ZEROES TEST
TEST DESCRIPTION:
THIS TEST WRITES AND READS A
SHIFTING ONE'S AND SHIFTING ZEROE'S
PATTERN TO ALL THE ML11'S
READ/WRITE REGISTERS

ROUTINES WRT REG AND RD REG
ACCEPT ARGUMENTS TO FURTHER
SELECT ROUTINES WHICH ACTUALLY
PERFORMS THE READING AND
WRITING OF THE REGISTERS.

THE DRIVE IS DROPPED ON DETECTED
ERRORS.

IMPLICIT INPUTS:
WT DATA
LOADED BY READ REGISTER ROUTINES AND
CONTAINS THE DATA PATTERN WRITTEN TO THE
REGISTERS (REPRESENTS GOOD DATA).

RD DATA
LOADED BY THE READ REGISTER ROUTINES AND
CONTAINS THE DATA PATTERN READ FROM THE
REGISTER (REPRESENTS BAD DATA).
!--
local
ERR_FLG,
TST_PAT,
index,
DODU_FLG:
CLR THRESHOLD;
DODD_FLG = ZERO;
TST_PAT = ONE;
incr SHIFT from 0 to 15 do
begin
incr TWICE from 0 to 1 do
begin

!ERROR FLAG PASSED TO ROUTINE
!TEST PATTERN
!POINTS TO REG PRESENTLY BEING TESTED
!DROP UNIT FLAG

!CLEAR ERROR PRINT THRESHOLD

!LOAD TST_PAT WITH A 1 IN A FILED OF 0'S

!DO SHIFT 16 TIMES

!REPEAT LOOP TWICE

```

9088 :ML4AD
9089 :
9090 : TEST CODE SECTION
9091 : 5055
9092 : 5056      incr REG_SEL from 0 to 7 do
9093 : 5057      begin
9094 : 5058      BGNSUB;
9095 : 5059      CLR_MBUS;
9096 : 5060      WRT_REG (.TST_PAT, .REG_SEL, index);      !WRITE TO THE REGISTER
9097 : 5061      RD_REG (.TST_PAT, .REG_SEL, ERR_FLG);      !READ THE REGISTER
9098 : 5062
9099 : 5063      if .ERR_FLG IS_SET
9100 : 5064      then
9101 : 5065      begin
9102 : 5066      CMP_THRESHOLD;      !IF THE ERROR FLAG IS SET THEN ERROR
9103 : 5067      !COMPARE ERROR PRINT THRESHOLD
9104 : 5068
9105 : 5069      selectone .REG_SEL of
9106 : 5070      set
9107 : 5071      [0 to 3] :
9108 : 5072      ERRDF (5, ASYNC, DUMPER);      !ASYNC MODULE FAILURE
9109 : 5073
9110 : 5074      [4 to 5] :
9111 : 5075      ERRDF (5, SYNC, DUMPER);      !SYNC MODULE FAILURE
9112 : 5076
9113 : 5077      [6 to 7] :
9114 : 5078      ERRDF (5, ARR_DAT, DUMPER);      !ARRAY DATA MODULE FAILURE
9115 : 5079
9116 : 5080      tes;
9117 : 5081      PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
9118 : 5082      PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
9119 : 5083      DODU_FLG = ONE;
9120 : 5084      end;
9121 : 5085
9122 : 5086      ENDSUB;
9123 : 5087      end;
9124 : 5088
9125 : 5089      TST_PAT = not .TST_PAT;
9126 : 5090      end;
9127 : 5091
9128 : 5092      TST_PAT = .TST_PAT^ONE;
9129 : 5093      end;
9130 : 5094
9131 : 5095      if .DODU_FLG IS_SET
9132 : 5096      then
9133 : 5097      begin
9134 : 5098      DODU (.ML_LUN);
9135 : 5099      DOCLN;
9136 : 5100      end;
9137 : 5101
9138 : 5102      ENDTST;
9142 :

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (58)

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

9256
9257
9258
9259 033470 104444
9260 033472 022626
9261 033474 000207
9262
9263
9264
9269
9270
9274
9275
9279 033476
9280 033476 004767 177320
9281 033502 104466
9282 033504 006000
9283 033506 103773
9284 033510 000207
9285
9286
9287
9292
9293
9294 : 5103 !
9295 : 5104 !<BLF/PAGE>

:ML4AD
:
TEST CODE SECTION
11\$: TRAP 44
CMP (SP)+,(SP)+
RTS PC ;
: Routine Size: 150 words
: Maximum stack depth per invocation: 25 words

.SBTTL T6 TEST CODE SECTION
T6::
1\$: JSR PC,\$T6 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

5001

5100

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (59)

9297 :ML4AD
9298 :
9299 :
9300 :
9301 :
9302 :
9303 :
9304 :
9305 :
9306 :
9307 :
9308 :
9309 :
9310 :
9311 :
9312 :
9313 :
9314 :
9315 :
9316 :
9317 :
9318 :
9319 :
9320 :
9321 :
9322 :
9323 :
9324 :
9325 :
9326 :
9327 :
9328 :
9329 :
9330 :
9331 :
9332 :
9333 :
9334 :
9335 :
9336 :
9337 :
9338 :
9339 :
9340 :
9341 :
9342 :
9343 :
9344 :
9345 :
9346 :
9347 :
9348 :
9349 :
9350 :
9351 :

5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129
5130
5131
5132
5133
5134
5135
5136
5137
5138
5139
5140
5141
5142
5143
5144
5145
5146
5147
5148
5149
5150
5151
5152
5153
5154
5155
5156

TEST CODE SECTION

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;

CLR_THRESHOLD;

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

!CLEAR ERROR PRINT THRESHOLD

```

9353 :ML4AD
9354 :
9355 :
9356 : 5157 ML_REG [19, FORCE_HI] = %o'177777';
9357 : 5158 DODU_FLG = ZERO;
9358 : 5159 TST_PAT = ONES;
9359 : 5160
9360 : 5161 incr TWICE from 0 to 1 do
9361 : 5162 begin
9362 : 5163
9363 : 5164 incr REG_SEL from 0 to 10 do
9364 : 5165 begin
9365 : 5166 BGNSUB;
9366 : 5167 CLR_MBUS;
9367 : 5168 WRT_REG (.TST_PAT, .REG_SEL, index);
9368 : 5169 CLR_DATA = (.RI) or (.IGNORE);
9369 : 5170 CLR_MBUS;
9370 : 5171 RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);
9371 : 5172
9372 : 5173 if .ERR_FLG IS_SET
9373 : 5174 then
9374 : 5175 begin
9375 : 5176 CMP_THRESHOLD;
9376 : 5177
9377 : 5178 selectone .REG_SEL of
9378 : 5179 set
9379 : 5180
9380 : 5181 [0, 1, 2, 3, 9, 10] :
9381 : 5182 ERRDF (6, ASYNC, DUMPER); !ASYNC MODULE FAILURE
9382 : 5183
9383 : 5184 [4, 5, 8] :
9384 : 5185 ERRDF (6, SYNC, DUMPER); !SYNC MODULE FAILURE
9385 : 5186
9386 : 5187 [6 to 7] :
9387 : 5188 ERRDF (6, ARR_DAT, DUMPER); !ARRAY DATA MODULE
9388 : 5189 tes;
9389 : 5190
9390 : 5191 PRINTB (FIV_FMT, PHR 4, WRD 12, WRD 52, FNC 23, WRD 56);
9391 : 5192 PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
9392 : 5193 DODU_FLG = ONE;
9393 : 5194 end;
9394 : 5195
9395 : 5196 ENDSUB;
9396 : 5197 end;
9397 : 5198
9398 : 5199 TST_PAT = not .TST_PAT;
9399 : 5200 end;
9400 : 5201
9401 : 5202 ML_REG [19, FORCE_HI] = ZEROES;
9402 : 5203
9403 : 5204
9404 : 5205 NOW TEST THE DRIVE TYPE REGISTER
9405 : 5206
9406 : 5207
9407 : 5208

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (59)

!CLR DATA FOR MLPD IS ONES
!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
!COMPARE ERROR PRINT THRESHOLD
!FIND WHICH MODULE FAILED
!ARRAY DATA MODULE
!REPEAT WITH COMPLIMENT BACKGROUND PAT
!RESTORE MLPD FORCED_HI

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (59)

```

9409 :ML4AD
9410 :
9411 : TEST CODE SECTION
9412 : 5209 if .MLDT neq .DRIVE_TYPE
9413 : 5210 then
9414 : 5211 begin
9415 : 5212 ERRDF (113, ASYNC, DUMPER);
9416 : 5213 PRINTB (TWO_FMT, REG 7, PHR 4);
9417 : 5214 PRINTB (FMT_2, .DRIVE_TYPE, .SAVE);
9418 : 5215 end;
9419 : 5216
9420 : 5217 if .DODU_FLG IS_SET
9421 : 5218 then
9422 : 5219 begin
9423 : 5220 DODU (.ML_LUN);
9424 : 5221 DOCLN;
9425 : 5222 end;
9426 : 5223
9427 : 5224 ENDTST;

```

!COMPARE REG CONTENTS TO CALCULATED VALUE

!DROP THIS UNIT IF DODU_FLG IS_SET

Address	OpCode	Operand 1	Operand 2	Operand 3	Comment	Address
9431					ENDTST;	
9432						
9436	033512	004167	151074		.SBTTL \$T7 TEST CODE SECTION	
9437	033516	162706	000006	\$T7:	JSR R1,\$\$SAVES	5102
9438	033522	005067	161650		SUB #6,SP	
9439	033526	012767	177777	162102	CLR P,CNT	
9440	033534	005005			MOV #-1,ML.REG+232	5154
9441	033536	012702	177777		CLR R5	5157
9442	033542	005001			MOV #-1,R2	DODU.FLG 5158
9443	033544	005004			CLR R1	*.TST.PAT 5159
9444	033546	104402		1\$:	CLR R4	TWICE 5161
9445	033550	152777	000040	2\$:	TRAP 2	REG.SEL 5164
9446	033556	016703	162250		BISB #40,@ML.REG+40	5165
9447	033562	042703	177770		MOV ML.DUT,R3	5166
9448	033566	142777	000007	161650	BIC #177770,R3	
9449	033574	150377	161644		BICB #7,@ML.REG+40	
9450	033600	010246			BISB R3,@ML.REG+40	
9451	033602	010446			MOV R2,-(SP)	: TST.PAT,* 5168
9452	033604	012746	000012		MOV R4,-(SP)	: REG.SEL,*
9453	033610	060616			MOV #12,-(SP)	
9454	033612	004767	171506		ADD SP,(SP)	: INDEX,*
9455	033616	016600	000012		JSR PC,WRT.REG	
9456	033622	006300			MOV 12(SP),R0	: INDEX,* 5169
9457	033624	006300			ASL R0	
9458	033626	006300			ASL R0	
9459	033630	010003			ASL R0	
9460	033632	016300	015406		MOV R0,R3	
9461	033636	056300	015412		MOV ML.REG+2(R3),R0	
9462	033642	010066	000006		BIS ML.REG+6(R3),R0	
9463	033646	152777	000040	161570	MOV R0,6(SP)	: *.CLR.DATA
					BISB #40,@ML.REG+40	

Address	OpCode	Op1	Op2	Op3	Instruction	Comments	Line
9465							
9466							
9467							
9468	033654	016700	162152		MOV ML,DUT,RO		
9469	033660	042700	177770		BIC #177770,RO		
9470	033664	142777	000007	161552	BICB #7,@ML,REG+40		
9471	033672	150077	161546		BISB RO,@ML,REG+40		
9472	033676	016616	000006		MOV 6(SP),(SP)	: CLR.DATA,*	
9473	033702	010446			MOV R4,-(SP)	: REG.SEL,*	5171
9474	033704	012746	000014		MOV #14,-(SP)		
9475	033710	060616			ADD SP,(SP)	: ERR.FLG,*	
9476	033712	004767	171774		JSR PC,RD,REG		
9477	033716	026627	000014	000001	CMP 14(SP),#1	: ERR.FLG,*	
9478	033724	001121			BNE 11\$		5173
9479	033726	005267	161444		INC P.CNT		
9480	033732	026767	161440	161440	CMP P.CNT,LIMIT		5175
9481	033740	003403			BLE 3\$		
9482	033742	062706	000012		ADD #12,SP		
9483	033746	000517			BR 13\$		
9484	033750	005704		3\$:	TST R4	: REG.SEL	
9485	033752	002403			BLT 4\$		5178
9486	033754	020427	000003		CMP R4,#3	: REG.SEL,*	
9487	033760	003406			BLE 5\$		
9488	033762	020427	000011	4\$:	CMP R4,#11	: REG.SEL,*	
9489	033766	002410			BLT 6\$		
9490	033770	020427	000012		CMP R4,#12	: REG.SEL,*	
9491	033774	003005			BGT 6\$		
9492	033776	104455		5\$:	TRAP 55		
9493	034000	000006			.WORD 6		5182
9494	034002	012706			.WORD ASYNC		
9495	034004	026302			.WORD DUMPER		
9496	034006	000430			BR 10\$		
9497	034010	020427	000004	6\$:	CMP R4,#4	: REG.SEL,*	5178
9498	034014	002403			BLT 7\$		
9499	034016	020427	000005		CMP R4,#5	: REG.SEL,*	
9500	034022	003403			BLE 8\$		
9501	034024	020427	000010	7\$:	CMP R4,#10	: REG.SEL,*	
9502	034030	001005			BNE 9\$		
9503	034032	104455		8\$:	TRAP 55		5185
9504	034034	000006			.WORD 6		
9505	034036	012750			.WORD SYNC		
9506	034040	026302			.WORD DUMPER		
9507	034042	000412			BR 10\$		
9508	034044	020427	000006	9\$:	CMP R4,#6	: REG.SEL,*	5178
9509	034050	002407			BLT 10\$		
9510	034052	020427	000007		CMP R4,#7	: REG.SEL,*	
9511	034056	003004			BGT 10\$		
9512	034060	104455			TRAP 55		
9513	034062	000006			.WORD 6		5188
9514	034064	013012			.WORD ARR.DAT		
9515	034066	026302			.WORD DUMPER		
9516	034070	012746	011344	10\$:	MOV #WRD.56,-(SP)		
9517	034074	012746	012464		MOV #FNC.23,-(SP)		5191
9518	034100	012746	011310		MOV #WRD.52,-(SP)		
9519	034104	012746	010650		MOV #WRD.12,-(SP)		

Address	OpCode	Operand1	Operand2	Comment	Label	Address
9521						
9522						
9523						
9524	034110	012746	011676	MOV #PHR.4,-(SP)		
9525	034114	012746	010304	MOV #FIV.FMT,-(SP)		
9526	034120	012746	000006	MOV #6,-(SP)		
9527	034124	010600		MOV SP,R0		
9528	034126	104414		TRAP 14		: SP,*
9529	034130	016716	161216	MOV RD.DATA,(SP)		
9530	034134	016646	000030	MOV 30(SP),-(SP)		: CLR.DATA,*
9531	034140	016346	015404	MOV ML.REG(R3),-(SP)		5192
9532	034144	012746	007436	MOV #FMT.16,-(SP)		
9533	034150	012746	000004	MOV #4,-(SP)		
9534	034154	010600		MOV SP,R0		: SP,*
9535	034156	104414		TRAP 14		
9536	034160	012705	000001	MOV #1,R5		: *.DODU.FLG
9537	034164	062706	000026	ADD #26,SP		5193
9538	034170	062706	000012	ADD #12,SP		5175
9539	034174	104467		TRAP 67		5165
9540	034176	006000		ROR R0		5194
9541	034200	103002		BHIS 13\$		
9542	034202	000167	177340	JMP 2\$		
9543	034206	005204		INC R4		
9544	034210	020427	000012	CMP R4,#12		: REG.SEL
9545	034214	003772		BLE 12\$: REG.SEL,*
9546	034216	005102		COM R2		
9547	034220	005201		INC R1		: TST.PAT
9548	034222	020127	000001	CMP R1,#1		: TWICE
9549	034226	003002		BGT 14\$: TWICE,*
9550	034230	000167	177310	JMP 1\$		
9551	034234	005067	161376	CLR ML.REG+232		
9552	034240	027767	161270	CMP @ML.REG+130,DRIVE.TYPE		: 5202
9553	034246	001431		BEQ 15\$: 5209
9554	034250	104455		TRAP 55		
9555	034252	000161		.WORD 161		: 5212
9556	034254	012706		.WORD ASYNC		
9557	034256	026302		.WORD DUMPER		
9558	034260	012746	011676	MOV #PHR.4,-(SP)		
9559	034264	012746	012546	MOV #REG.7,-(SP)		: 5213
9560	034270	012746	010246	MOV #TWO.FMT,-(SP)		
9561	034274	012746	000003	MOV #3,-(SP)		
9562	034300	010600		MOV SP,R0		: SP,*
9563	034302	104414		TRAP 14		
9564	034304	010416		MOV R4,(SP)		: SAVE,*
9565	034306	016746	161042	MOV DRIVE.TYPE,-(SP)		5214
9566	034312	012746	006506	MOV #FMT.2,-(SP)		
9567	034316	012746	000003	MOV #3,-(SP)		
9568	034322	010600		MOV SP,R0		: SP,*
9569	034324	104414		TRAP 14		
9570	034326	062706	000016	ADD #16,SP		
9571	034332	005305		DEC R5		: DODU.FLG
9572	034334	001004		BNE 16\$		5217
9573	034336	016700	161466	MOV ML.LUN,R0		
9574	034342	104451		TRAP 51		: 5220
9575	034344	104444		TRAP 44		

9577
9578
9579
9580 034346 062706 000006
9581 034352 000207
9582
9583
9584
9589
9590
9594
9595
9599 034354
9600 034354 004767 177132
9601 034360 104466
9602 034362 006000
9603 034364 103773
9604 034366 000207
9605
9606
9607
9612
9613
9614 :

:ML4AD
:
TEST CODE SECTION
16\$: ADD #6,SP
RTS PC
:
: Routine Size: 209 words
: Maximum stack depth per invocation: 25 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<
5102

.SBTTL T7 TEST CODE SECTION
T7::
1\$: JSR PC,\$T7
TRAP 66
ROR R0
BLO 1\$
RTS PC
:
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

5222

5225 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (60)

```

9616 :ML4AD
9617 :
9618 :
9619 : 5226
9620 : 5227
9621 : 5228
9622 : 5229
9623 : 5230
9624 : 5231
9625 : 5232
9626 : 5233
9627 : 5234
9628 : 5235
9629 : 5236
9630 : 5237
9631 : 5238
9632 : 5239
9633 : 5240
9634 : 5241
9635 : 5242
9636 : 5243
9637 : 5244
9638 : 5245
9639 : 5246
9640 : 5247
9641 : 5248
9642 : 5249
9643 : 5250
9644 : 5251
9645 : 5252
9646 : 5253
9647 : 5254
9648 : 5255
9649 : 5256
9650 : 5257
9651 : 5258
9652 : 5259
9653 : 5260
9654 : 5261
9655 : 5262
9656 : 5263
9657 : 5264
9658 : 5265
9659 : 5266
9660 : 5267
9661 : 5268
9662 : 5269
9663 : 5270
9664 : 5271
9665 : 5272
9666 : 5273
9667 : 5274
9668 : 5275
9669 : 5276
9670 : 5277

```

```

TEST CODE SECTION
BGNTST;
++
TEST NUMBER: TST 8
TEST NAME: REGISTER SELECTION TEST
TEST DESCRIPTION:
THIS TEST TESTS FOR UNIQUE REGISTER
SELECTION BY FIRST WRITING A BACKGROUND
PATTERN INTO ALL READ/WRITE REGISTERS
IT THEN WRITES A COMPLIMENT
BACKGROUND PATTERN INTO ONE REGISTER
AND READS THE REMAINING UNWRITTEN
REGISTERS FOR AN UNCHANGED BACKGROUND
PAT
THIS PROCEDURE IS REPEATED UNTIL ALL
REGISTERS HAVE BEEN WRITTEN WITH A
COMPLIMENT BACKGROUND PATTERN.

```

--

local

```

DODU_FLG,
TST_PAT,
ERR_FLG,
index;

```

CLR THRESHOLD;

BGNSUB;

CLR MBUS;

DODU_FLG = ZERO;

TST_PAT = %0'125252';

WRT_CS1 (.TST_PAT, 0);

WRT_ER (.TST_PAT, 6);

WRT_DA (.TST_PAT, 3);

WRT_PA (.TST_PAT, 8);

WRT_E1 (.TST_PAT, 13);

WRT_E2 (.TST_PAT, 14);

incr CNT_1 from 0 to 4 do

begin

TST_PAT = not .TST_PAT;

case .CNT_1 from 0 to 4 of

set

```

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

```

!CLEAR ERROR PRINT THRESHOLD

```

!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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (60)

```

9672 :ML4AD
9673 :
9674 :
9675 : 5278
9676 : 5279
9677 : 5280
9678 : 5281
9679 : 5282
9680 : 5283
9681 : 5284
9682 : 5285
9683 : 5286
9684 : 5287
9685 : 5288
9686 : 5289
9687 : 5290
9688 : 5291
9689 : 5292
9690 : 5293
9691 : 5294
9692 : 5295
9693 : 5296
9694 : 5297
9695 : 5298
9696 : 5299
9697 : 5300
9698 : 5301
9699 : 5302
9700 : 5303
9701 : 5304
9702 : 5305
9703 : 5306
9704 : 5307
9705 : 5308
9706 : 5309
9707 : 5310
9708 : 5311
9709 : 5312
9710 : 5313
9711 : 5314
9712 : 5315
9713 : 5316
9714 : 5317
9715 : 5318
9716 : 5319
9717 : 5320
9718 : 5321
9719 : 5322
9720 : 5323
9721 : 5324
9722 : 5325
9723 : 5326
9724 : 5327
9725 : 5328
9726 : 5329

TEST CODE SECTION

[0] :
WRT_CS1 (.TST_PAT, 0); !FIRST PASS WRITE COMP PAT TO MLCS1

[1] :
WRT_ER (.TST_PAT, 6); !SECOND PASS WRITE COMP PAT TO MLER

[2] :
WRT_DA (.TST_PAT, 3); !THIRD PASS WRITE COMP PAT TO MLDA

[3] :
WRT_PA (.TST_PAT, 8); !FORTH PASS WRITE COMP PAT TO MLPA

[4] :
WRT_E1 (.TST_PAT, 13); !FIFTH PASS WRITE COMP PAT TO MLE1
tes:

TST_PAT = not .TST_PAT; !COMPLIMENT TST_PAT BACK TO BACKGROUND

incr CNT_2 from .CNT_1 + 1 to 5 do
begin
!NOW READ THE REMAINING UNWRITTEN REGISTERS FOR AN
!UNCHANGED BACKGROUND

case .CNT_2 from 0 to 5 of
set
!SELECT THE REGISTER TO READ

[0] :
RD_CS1 (.TST_PAT, index = 0, ERR_FLG); !READ MLCS1

[1] :
RD_ER (.TST_PAT, index = 6, ERR_FLG); !READ MLER

[2] :
RD_DA (.TST_PAT, index = 3, ERR_FLG); !READ MLDA

[3] :
RD_PA (.TST_PAT, index = 8, ERR_FLG); !READ MLPA

[4] :
RD_E1 (.TST_PAT, index = 13, ERR_FLG); !READ MLE1

[5] :
RD_E2 (.TST_PAT, index = 14, ERR_FLG); !READ MLE2
tes:

if .ERR_FLG IS_SET
then
!SEE IF READ FOUND AN ERROR
begin
CMP_THRESHOLD;
!COMPARE ERROR PRINT THRESHOLD

selectone .CNT_2 of
set

[0 to 2] :

```

```

9728 :ML4AD
9729 :
9730 : TEST CODE SECTION
9731 : 5330 ERRDF (110, ASYNC, DUMPER); !ASYNC MODULE FAILURE
9732 : 5331
9733 : 5332 [3 to 5]
9734 : 5333 ERRDF (110, ARR_DAT, DUMPER); !ARRAY DATA MODULE FAILURE
9735 : 5334 tes:
9736 : 5335
9737 : 5336 PRINTB (THR_FMT, WRD 38, WRD 37, WRD 10);
9738 : 5337 PRINTB (FMT-16, .ML_REG [.index, REGISTER_ADD], .WT_DATA, .RD_DATA);
9739 : 5338 DODU_FLG = ONE;
9740 : 5339 end;
9741 : 5340
9742 : 5341 end;
9743 : 5342
9744 : 5343 end;
9745 : 5344
9746 : 5345 ENDSUB;
9747 : 5346
9748 : 5347 if .DODU_FLG IS_SET !DROP THIS UNIT IF DODU_FLG IS_SET
9749 : 5348 then
9750 : 5349 begin
9751 : 5350 DODU (.ML_LUN);
9752 : 5351 DOCLN;
9753 : 5352 end;
9754 : 5353
9755 : 5354 ENDTST;
9759 :
9760 :
  
```

Address	Hex	Dec	Hex	Label	Instruction	Comment	Address
9764	034370	004167	150216	\$T8:	.SBTTL	\$T8 TEST CODE SECTION	
9765	034374	005746			JSR	R1,\$SAVE5	5224
9766	034376	005067	160774		TST	-(SP)	
9767	034402	104402		1\$:	CLR	P.CNT	5257
9768	034404	152777	000040	161032	TRAP	2	5259
9769	034412	016703	161414		BISB	#40,.@ML.REG+40	5260
9770	034416	042703	177770		MOV	ML,DUT,R3	
9771	034422	142777	000007	161014	BIC	#177770,R3	
9772	034430	150377	161010		BICB	#7,.@ML.REG+40	
9773	034434	005005			BISB	R3,.@ML.REG+40	
9774	034436	012704	125252		CLR	R5	: DODU.FLG 5262
9775	034442	010446			MOV	#-52526,R4	: *.TST.PAT 5263
9776	034444	005046			MOV	R4,-(SP)	: TST.PAT,* 5264
9777	034446	004707	165240		CLR	-(SP)	
9778	034452	010416			JSR	PC,WRT,CS1	
9779	034454	012746	000006		MOV	R4,(SP)	: TST.PAT,* 5265
9780	034460	004767	165404		MOV	#6,-(SP)	
9781	034464	010416			JSR	PC,WRT,ER	
9782	034466	012746	000003		MOV	R4,(SP)	: TST.PAT,* 5266
					MOV	#3,-(SP)	

Address	OpCode	Op2	Op3	Op4	Instruction	Comments	Line
9784							
9785							
9786							
9787	034472	004767	165550		JSR PC,WRT.DA		
9788	034476	010416			MOV R4,(SP)		
9789	034500	012746	000010		MOV #10,-(SP)	: TST.PAT,*	5267
9790	034504	004767	166072		JSR PC,WRT.PA		
9791	034510	010416			MOV R4,(SP)	: T.PAT,*	5268
9792	034512	012746	000015		MOV #15,-(SP)		
9793	034516	004767	166266		JSR PC,WRT.E1		
9794	034522	010416			MOV R4,(SP)	: TST.PAT,*	5269
9795	034524	012746	000016		MOV #16,-(SP)		
9796	034530	004767	166462		JSR PC,WRT.E2		
9797	034534	005001			CLR R1	: CNT.1	5271
9798	034536	005104			COM R4	: TST.PAT	5273
9799	034540	010103			MOV R1,R3	: CNT.1,*	5275
9800	034542	006303			ASL R3		
9801	034544	066307	034550		ADD 3\$(R3),PC		
9802	034550	000012			.WORD 4\$-3\$		
9803	034552	000024			.WORD 5\$-3\$		
9804	034554	000040			.WORD 6\$-3\$		
9805	034556	000054			.WORD 7\$-3\$		
9806	034560	000070			.WORD 8\$-3\$		
9807	034562	010446			MOV R4,-(SP)	: TST.PAT,*	5279
9808	034564	005046			CLR -(SP)		
9809	034566	004767	165120		JSR PC,WRT.CS1		
9810	034572	000427			BR 9\$		
9811	034574	010446			MOV R4,-(SP)	: TST.PAT,*	5275
9812	034576	012746	000006		MOV #6,-(SP)	: TST.PAT,*	5282
9813	034602	004767	165262		JSR PC,WRT.ER		
9814	034606	000421			BR 9\$		
9815	034610	010446			MOV R4,-(SP)	: TST.PAT,*	5275
9816	034612	012746	000003		MOV #3,-(SP)	: TST.PAT,*	5285
9817	034616	004767	165424		JSR PC,WRT.DA		
9818	034622	000413			BR 9\$		
9819	034624	010446			MOV R4,-(SP)	: TST.PAT,*	5275
9820	034626	012746	000010		MOV #10,-(SP)	: TST.PAT,*	5288
9821	034632	004767	165744		JSR PC,WRT.PA		
9822	034636	000405			BR 9\$		
9823	034640	010446			MOV R4,-(SP)	: TST.PAT,*	5275
9824	034642	012746	000015		MOV #15,-(SP)	: TST.PAT,*	5291
9825	034646	004767	166136		JSR PC,WRT.E1		
9826	034652	005104			COM R4	: TST.PAT	5294
9827	034654	010103			MOV R1,R3	: CNT.1,CNT.2	5296
9828	034656	000167	000416		JMP 23\$		
9829	034662	010300			MOV R3,R0	: CNT.2,*	5299
9830	034664	006300			ASL R0		
9831	034666	066007	034672		ADD 11\$(R0),PC		
9832	034672	000014			.WORD 12\$-11\$		
9833	034674	000036			.WORD 13\$-11\$		
9834	034676	000062			.WORD 14\$-11\$		
9835	034700	000106			.WORD 15\$-11\$		
9836	034702	000132			.WORD 16\$-11\$		
9837	034704	000156			.WORD 17\$-11\$		
9838	034706	010446			MOV R4,-(SP)	: TST.PAT,*	5303

Address	OpCode	Op2	Op3	Op4	Label	Comment	Seq
9840					:ML4AD		
9841					:	TEST CODE SECTION	
9842							
9843	034710	005002				CLR R2	
9844	034712	005046				CLR -(SP)	: INDEX
9845	034714	012746	000030			MOV #30, -(SP)	
9846	034720	060616				ADD SP, (SP)	: ERR.FLG,*
9847	034722	004767	165034			JSR PC, RD.CS1	
9848	034726	000461				BR 18\$	
9849	034730	010446		13\$:		MOV R4, -(SP)	: TST.PAT,*
9850	034732	012702	000006			MOV #6, R2	: * ,INDEX
9851	034736	010246				MOV R2, -(SP)	: INDEX,*
9852	034740	012746	000030			MOV #30, -(SP)	
9853	034744	060616				ADD SP, (SP)	: ERR.FLG,*
9854	034746	004767	165166			JSR PC, RD.ER	
9855	034752	000447				BR 18\$	
9856	034754	010446		14\$:		MOV R4, -(SP)	: TST.PAT,*
9857	034756	012702	000003			MOV #3, R2	: * ,INDEX
9858	034762	010246				MOV R2, -(SP)	: INDEX,*
9859	034764	012746	000030			MOV #30, -(SP)	
9860	034770	060616				ADD SP, (SP)	: ERR.FLG,*
9861	034772	004767	165320			JSR PC, RD.DA	
9862	034776	000435				BR 18\$	
9863	035000	010446		15\$:		MOV R4, -(SP)	: TST.PAT,*
9864	035002	012702	000010			MOV #10, R2	: * ,INDEX
9865	035006	010246				MOV R2, -(SP)	: INDEX,*
9866	035010	012746	000030			MOV #30, -(SP)	
9867	035014	060616				ADD SP, (SP)	: ERR.FLG,*
9868	035016	004767	165644			JSR PC, RD.PA	
9869	035022	000423				BR 18\$	
9870	035024	010446		16\$:		MOV R4, -(SP)	: TST.PAT,*
9871	035026	012702	000015			MOV #15, R2	: * ,INDEX
9872	035032	010246				MOV R2, -(SP)	: INDEX,*
9873	035034	012746	000030			MOV #30, -(SP)	
9874	035040	060616				ADD SP, (SP)	: ERR.FLG,*
9875	035042	004767	166026			JSR PC, RD.E1	
9876	035046	000411				BR 18\$	
9877	035050	010446		17\$:		MOV R4, -(SP)	: TST.PAT,*
9878	035052	012702	000016			MOV #16, R2	: * ,INDEX
9879	035056	010246				MOV R2, -(SP)	: INDEX,*
9880	035060	012746	000030			MOV #30, -(SP)	
9881	035064	060616				ADD SP, (SP)	: ERR.FLG,*
9882	035066	004767	166222			JSR PC, RD.E2	
9883	035072	026627	000030	000001	18\$:	CMP 30(SP), #1	: ERR.FLG,*
9884	035100	001075				BNE 22\$	
9885	035102	005267	160270			INC P.CNT	
9886	035106	026767	160264	160264		CMP P.CNT, LIMIT	: 5323
9887	035114	003403				BLE 19\$	
9888	035116	062706	000006			ADD #6, SP	
9889	035122	000474				BR 24\$	
9890	035124	005703		19\$:		TST R3	: CNT.2
9891	035126	002410				BLT 20\$	
9892	035130	020327	000002			CMP R3, #2	: CNT.2,*
9893	035134	003005				BGT 20\$	
9894	035136	104455				TRAP 55	: 5330

Address	Code	Label	Instruction	Comments	Page
9896					
9897					
9898					
9899	035140	000156	.WORD	156	
9900	035142	012706	.WORD	ASYNC	
9901	035144	026302	.WORD	DUMPER	
9902	035146	000412	BR	21\$	
9903	035150	020327	000003	20\$: CMP R3,#3	: CNT.2,* 5326
9904	035154	002407	BLT	21\$	
9905	035156	020327	000005	20\$: CMP R3,#5	: CNT.2,*
9906	035162	003004	BGT	21\$	
9907	035164	104455	TRAP	55	
9908	035166	000156	.WORD	156	5333
9909	035170	013012	.WORD	APP.DAT	
9910	035172	026302	.WORD	DUMPER	
9911	035174	012746	010630	21\$: MOV #WRD.10,-(SP)	: 5336
9912	035200	012746	011122	MOV #WRD.37,-(SP)	
9913	035204	012746	011132	MOV #WRD.38,-(SP)	
9914	035210	012746	010256	MOV #THR.FMT,-(SP)	
9915	035214	012746	000004	MOV #4,-(SP)	
9916	035220	010600	MOV	SP,R0	: SP,*
9917	035222	104414	TRAP	14	
9918	035224	016716	160122	MOV RD.DATA,(SP)	
9919	035230	016746	160114	MOV WT.DATA,-(SP)	5337
9920	035234	010200	MOV	R2,R0	: INDEX,*
9921	035236	006300	ASL	R0	
9922	035240	006300	ASL	R0	
9923	035242	006300	ASL	R0	
9924	035244	016046	015404	MOV ML.REG(R0),-(SP)	
9925	035250	012746	007436	MOV #FMT.16,-(SP)	
9926	035254	012746	000004	MOV #4,-(SP)	
9927	035260	010600	MOV	SP,R0	: SP,*
9928	035262	104414	TRAP	14	
9929	035264	012705	000001	MOV #1,R5	: *,DODU.FLG
9930	035270	062706	000022	ADD #22,SP	5338
9931	035274	062706	000006	ADD #6,SP	5323
9932	035300	005203	23\$: INC R3	: CNT.2	5297
9933	035302	020327	000005	23\$: CMP R3,#5	: CNT.2,* 5296
9934	035306	003002	BGT	24\$	
9935	035310	000167	177346	JMP 10\$	
9936	035314	022626	24\$: CMP (SP)+,(SP)+	: CNT.1	5272
9937	035316	005201	INC	R1	: CNT.1,* 5271
9938	035320	020127	000004	CMP R1,#4	
9939	035324	003002	BGT	25\$	
9940	035326	000167	177204	JMP 2\$	
9941	035332	062706	000016	25\$: ADD #16,SP	: 5259
9942	035336	104467	TRAP	67	5343
9943	035340	006000	ROR	R0	
9944	035342	103002	BHIS	26\$	
9945	035344	000167	177032	JMP 1\$	
9946	035350	005305	26\$: DEC R5	: DODU.FLG	5347
9947	035352	001004	BNE	27\$	
9948	035354	016700	160450	MOV ML.LUN,R0	: 5350
9949	035360	104451	TRAP	51	
9950	035362	104444	TRAP	44	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

9952
9953
9954
9955 035364 005726
9956 035366 000207
9957
9958
9959
9964
9965
9969
9970
9974 035370
9975 035370 004767 176774
9976 035374 104466
9977 035376 006000
9978 035400 103773
9979 035402 000207
9980
9981
9982
9987
9988
9989 :

:ML4AD
:
TEST CODE SECTION
27\$: TST (SP)+
RTS PC ;
: Routine Size: 256 words
: Maximum stack depth per invocation: 28 words

.SBTTL T8 TEST CODE SECTION
T8::
1\$: JSR PC,ST8 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

5352

5355 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (61)

```

9991 :ML4AD
9992 :
9993 : TEST CODE SECTION
9994 : 5356 :
9995 : 5357 :
9996 : 5358 BGNTST:
9997 : 5359 :
9998 : 5360 :++
9999 : 5361 : TEST NUMBER: TST 9
10000 : 5362 :
10001 : 5363 : TEST NAME: PRINT SERIAL NUMBER
10002 : 5364 :
10003 : 5365 : TEST DESCRIPTION:
10004 : 5366 : UPON A YES RESPONSE TO THE
10005 : 5367 : SOFTWARE QUESTION 'PRINT SERIAL NO?
10006 : 5368 : PRINT OUT THE DRIVE UNDER TEST
10007 : 5369 : SERIAL NUMBER.
10008 : 5370 :
10009 : 5371 :--
10010 : 5372 :
10011 : 5373 Local
10012 : 5374 D3,
10013 : 5375 D2,
10014 : 5376 D1,
10015 : 5377 D0:
10016 : 5378
10017 : 5379 if .PRSN IS_SET
10018 : 5380 then
10019 : 5381 begin
10020 : 5382 D3 = .SN3;
10021 : 5383 D2 = .SN2;
10022 : 5384 D1 = .SN1;
10023 : 5385 D0 = .SN0;
10024 : 5386
10025 : 5387 if ((.D3 gtr 9) or (.D2 gtr 9) or (.D1 gtr 9) or (.D0 gtr 9))
10026 : 5388 then
10027 : 5389 !SEE IF DIGITS ARE TO BIG FOR COVERSION
10028 : 5390 PRINTB (FMT_25, PHR_15, .MLSN) !PRINT OCTAL SN IF TO BIG
10029 : 5391 else
10030 : 5392 PRINTB (FMT_26, PHR_15, .D3, .D2, .D1, .D0); !ELSE PRINT DECIMAL SERIAL NUMBER
10031 : 5393
10032 : 5394 end;
10033 : 5395
10034 : 5396 ENDTST:

```

```

!STORES DIGIT 3 OF SERIAL NUMBER
!STORES DIGIT 2 OF SERIAL NUMBER
!STORES DIGIT 1 OF SERIAL NUMBER
!STORES DIGIT 0 OF SERIAL NUMBER
!SEE IF WE PRINT THE SERIAL NUMBER
!PRINT THE SERIAL NUMBER IF THE REPLY WAS YES
!LOAD DIGIT 3 OF SN INTO D3
!LOAD DIGIT 2 OF SN INTO D2
!LOAD DIGIT 1 OF SN INTO D1
!LOAD DIGIT 0 OF SN INTO D0

```

```

10039 :
10043 035404 004167 147144 :.SBTTL $T9 TEST CODE SECTION
10044 035410 026727 144764 000001 JSR R1,$SAVE3
10045 035416 001101 BNE 4$

```

5354
5379

Address	OpCode	Op2	Op3	Op4	Instruction	Register	Value
10047							
10048							
10049							
10050	035420	017703	160120		MOV	@ML.REG+140,R3	
10051	035424	006203			ASR	R3	*D3
10052	035426	006203			ASR	R3	D3
10053	035430	006203			ASR	R3	D3
10054	035432	006203			ASR	R3	D3
10055	035434	000303			SWAB	R3	D3
10056	035436	042703	177760		BIC	#177760,R3	*D3
10057	035442	017700	160076		MOV	@ML.REG+140,R0	*D2
10058	035446	000300			SWAB	R0	D2
10059	035450	042700	177760		BIC	#177760,R0	*D2
10060	035454	117701	160064		MOVB	@ML.REG+140,R1	*D1
10061	035460	006201			ASR	R1	D1
10062	035462	006201			ASR	R1	D1
10063	035464	006201			ASR	R1	D1
10064	035466	006201			ASR	R1	D1
10065	035470	042701	177760		BIC	#177760,R1	*D1
10066	035474	117702	160044		MOVB	@ML.REG+140,R2	*D0
10067	035500	042702	177760		BIC	#177760,R2	*D0
10068	035504	020327	000011		CMP	R3,#11	D3,*
10069	035510	003011			BGT	1\$	
10070	035512	020027	000011		CMP	R0,#11	D2,*
10071	035516	003006			BGT	1\$	
10072	035520	020127	000011		CMP	R1,#11	D1,*
10073	035524	003003			BGT	1\$	
10074	035526	020227	000011		CMP	R2,#11	D0,*
10075	035532	003413			BLE	2\$	
10076	035534	017746	160004	1\$:	MOV	@ML.REG+140,-(SP)	
10077	035540	012746	012136		MOV	#PHR.15,-(SP)	
10078	035544	012746	010166		MOV	#FMT.25,-(SP)	
10079	035550	012746	000003		MOV	#3,-(SP)	
10080	035554	010600			MOV	SP,R0	SP,*
10081	035556	104414			TRAP	14	
10082	035560	000416			BR	3\$	
10083	035562	010246		2\$:	MOV	R2,-(SP)	D0,*
10084	035564	010146			MOV	R1,-(SP)	D1,*
10085	035566	010046			MOV	R0,-(SP)	D2,*
10086	035570	010346			MOV	R3,-(SP)	D3,*
10087	035572	012746	012136		MOV	#PHR.15,-(SP)	
10088	035576	012746	010206		MOV	#FMT.26,-(SP)	
10089	035602	012746	000006		MOV	#6,-(SP)	
10090	035606	010600			MOV	SP,R0	SP,*
10091	035610	104414			TRAP	14	
10092	035612	062706	000006		ADD	#6,SP	
10093	035616	062706	000010	3\$:	ADD	#10,SP	
10094	035622	000207		4\$:	RTS	PC	
10095							
10096							
10097							

: Routine Size: 72 words
 : Maximum stack depth per invocation: 11 words

10103 ;ML4AD

10104 :

10105

10109

10110

10114 035624

10115 035624 004767 177554

10116 035630 104466

10117 035632 006000

10118 035634 103773

10119 035636 000207

10120

10121

10122

10127

10128

10129 : 5397 !<BLF/PAGE>

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (61)

.SBTTL T9 TEST CODE SECTION

T9::

1\$:

JSR PC,\$T9

TRAP 66

ROR R0

BLO 1\$

RTS PC

5394

; Routine Size: 6 words

; Maximum stack depth per invocation: 0 words

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (62)

10131 :ML4AD
10132 :
10133 :
10134 :
10135 :
10136 :
10137 :
10138 :
10139 :
10140 :
10141 :
10142 :
10143 :
10144 :
10145 :
10146 :
10147 :
10148 :
10149 :
10150 :
10151 :
10152 :
10153 :
10154 :
10155 :
10156 :
10157 :
10158 :
10159 :
10160 :
10161 :
10162 :
10163 :
10164 :
10165 :
10166 :
10167 :
10168 :
10169 :
10170 :
10171 :
10172 :
10173 :
10174 :
10175 :
10176 :
10177 :
10178 :
10179 :
10180 :
10181 :
10182 :
10183 :
10184 :
10185 :

5398
5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 10

TEST NAME: C-BUS PARITY TEST

TEST DESCRIPTION:

TEST THE CONTROL BUS PARITY
DETECTION AND GENERATING BY:

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

--

local

SAVE,
TST_PAT;

!TEMPORARY SAVE LOCATION
!TEST PATTERN

if .PAR_DIS IS_NOT_SET
then

!SEE IF PARITY IS DISABLED

begin
TST_PAT = %0'125252':

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

incr TWICE from 1 to 2 do

!REPEAT LOOP TWICE

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

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

if .CPAR IS_NOT_SET
then

!READ CPAR BIT SET

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

!ERROR IF NOT SET

CLR_MBUS;
MLDA = .TST_PAT;

!CLEAR OUT PAT BIT
!WRITE ODD PARITY CONTROL BUS

if .CPAR IS_SET
then

!READ CPAR BIT CLEARED

10299
10300
10301
10302 036244 012746 010246
10303 036250 012746 000003
10304 036254 010600
10305 036256 104414
10306 036260 062706 000010
10307 036264 000207
10308
10309
10310
10315
10316
10320
10321
10325 036266
10326 036266 004767 177346
10327 036272 104466
10328 036274 006000
10329 036276 103773
10330 036300 000207
10331
10332
10333
10338
10339
10340 ;

:ML4AD
:
TEST CODE SECTION
MOV #TWO.FMT,-(SP)
MOV #3,-(SP)
MOV SP,R0 : SP,*
TRAP 14
ADD #10,SP
RTS PC
:
: Routine Size: 139 words
: Maximum stack depth per invocation: 9 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

5396

.SBTTL T10 TEST CODE SECTION
T10::
1\$: JSR PC,\$T10 :
TRAP 66
ROR R0
BLO 1\$
RTS PC
:
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

5470

5473 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<N:ALE>ML4AD.BLI.4 (63)

10342 :ML4AD

10343 :
10344 :
10345 : 5474
10346 : 5475
10347 : 5476
10348 : 5477
10349 : 5478
10350 : 5479
10351 : 5480
10352 : 5481
10353 : 5482
10354 : 5483
10355 : 5484
10356 : 5485
10357 : 5486
10358 : 5487
10359 : 5488
10360 : 5489
10361 : 5490
10362 : 5491
10363 : 5492
10364 : 5493
10365 : 5494
10366 : 5495
10367 : 5496
10368 : 5497
10369 : 5498
10370 : 5499
10371 : 5500
10372 : 5501
10373 : 5502
10374 : 5503
10375 : 5504
10376 : 5505
10377 : 5506
10378 : 5507
10379 : 5508
10380 : 5509
10381 : 5510
10382 : 5511
10383 : 5512
10384 : 5513
10385 : 5514
10389 :
10390 :
10394 036302 004167 146232
10395 036306 104402
10396 036310 152777 000040 157126

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 11

TEST NAME: MEMORY SIZING TEST

TEST DESCRIPTION:

THIS TESTS THE ML11'S SIZING
LOGIC BY COMPARING THE
OPERATORS INPUTED NUMBER OF ARRAYS
PRESENT TO THE ML11 SIZING
NUMBER OF ARRAYS PRESENT
THE DRIVE IS DROPPED ON DETECTED ERRORS.

IMPLICIT INPUTS:

OF NUM_ARR:
LOADED DURING INIT CODE AND
STORES OPERATORS INPUT TO THE
SOFTWARE QUESTION:
NUMBER OF ARRAYS PRESENT?

!--

BGNSUB;

CLR_MBUS;

if (.OP_NUM_ARR + 1) neq .ML_NUM_ARR
then

!SEE IF DRIVE SIZED SAME NO. OF ARRAYS AS OP INPUTED

begin

ERRDF (10, ASYNC, DUMPER);

!IF NOT EQL THEN ERROR AND SET DODU_FLG

PRINTB (TWO_FMT, FNC 1, WRD 14);

PRINTB (FMT_2, (.OP_NUM_ARR + 1), .ML_NUM_ARR);

end;

ENDSUB;

ENDTST;

.SBTTL \$T11 TEST CODE SECTION
JSR R1,\$SAVE2
TRAP 2
BISB #40,@ML.REG+40

5472
5476
5502

10459
10460
10464 036510
10465 036510 004767 177566
10466 036514 104466
10467 036516 006000
10468 036520 103773
10469 036522 000207
10470
10471
10472
10477
10478
10479 :

T11:: .SBTTL T11 TEST CODE SECTION

1\$: JSR PC,\$T11
TRAP 66
ROR R0
BLO 1\$
RTS PC

5513

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

5515 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (64)

```

10537 :ML4AD
10538 :
10539 :
10540 :      5568      end:
10541 :      5569
10542 :      5570      ENDSUB:
10543 :      5571      ENDTST:
10547 :
10548 :

```

```

10552 036524 010146      $T12:  .SBTTL  $T12 TEST CODE SECTION
10553 036526 005746      MOV    R1,-(SP)
10554 036530 104402      TST   -(SP)
10555 036532 152777 000040 156704 1$:  TRAP  2
10556 036540 016701 157266      BISB  #40,@ML.REG+40
10557 036544 042701 177770      MOV   ML.DUT,R1
10558 036550 142777 000007 156666      BIC   #177770,R1
10559 036556 150177 156662      BICB  #7,@ML.REG+40
10560 036562 012777 000001 156614      BISB  R1,@ML.REG+40
10561 036570 012700 000001      MOV   #1,@ML.REG
10562 036574 001410      MOV   #1,R0
10563 036576 016701 143314 2$:  BEQ   5$
10564 036602 001403      MOV   LSDLY,R1
10565 036604 005016      BEQ   4$
10566 036606 005301 3$:  CLR   (SP)
10567 036610 001375      DEC   R1
10568 036612 005300 4$:  BNE   3$
10569 036614 000767      DEC   R0
10570 036616 132777 000001 156560 5$:  BR    2$
10571 036624 001426      BITB  #1,@ML.REG
10572 036626 104455      BEQ   6$
10573 036630 000013      TRAP  55
10574 036632 012706      .WORD 13
10575 036634 026302      .WORD ASYNC
10576 036636 012746 010730      .WORD DUMPER
10577 036642 012746 012164      MOV   #WRD.19,-(SP)
10578 036646 012746 010640      MOV   #FNC.2,-(SP)
10579 036652 012746 011626      MOV   #WRD.11,-(SP)
10580 036656 012746 010520      MOV   #PHR.2,-(SP)
10581 036662 012746 010304      MOV   #WRD.1,-(SP)
10582 036666 012746 000006      MOV   #FIV.FMT,-(SP)
10583 036672 010600      MOV   #6,-(SP)
10584 036674 104414      MOV   SP,R0
10585 036676 062706 000016      TRAP  14
10586 036702 132777 000001 156554 6$:  ADD   #16,SP
10587 036710 001426      BITB  #1,@ML.REG+60
10588 036712 104455      BEQ   7$
10589 036714 000014      TRAP  55
10590 036716 012706      .WORD 14
10591 036720 026302      .WORD ASYNC
                          .WORD DUMPER

```

5514
5518
5544
5546
5547
5549
5552
5553
5551
5556
5559

10649
10650
10651
10652 037074 006000
10653 037076 103773
10654 037100 000207
10655
10656
10657
10662
10663
10664 ;

;ML4AD
;

TEST CODE SECTION

ROR R0
BLO 1\$
RTS PC

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

5572 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (65)

10666 :ML4AD
 10667 .
 10668 .
 10669 :
 10670 :
 10671 :
 10672 :
 10673 :
 10674 :
 10675 :
 10676 :
 10677 :
 10678 :
 10679 :
 10680 :
 10681 :
 10682 :
 10683 :
 10684 :
 10685 :
 10686 :
 10687 :
 10688 :
 10689 :
 10690 :
 10691 :
 10692 :
 10693 :
 10694 :
 10695 :
 10696 :
 10697 :
 10698 :
 10699 :
 10700 :
 10701 :
 10702 :
 10703 :
 10704 :
 10705 :
 10706 :
 10707 :
 10708 :
 10709 :
 10710 :
 10711 :
 10712 :
 10713 :
 10714 :
 10715 :
 10716 :
 10717 :
 10718 :
 10719 :
 10720 :

```

TEST CODE SECTION
BGNTST:
!++
TEST NUMBER: TST 13
TEST NAME: WRITE CHECK FUNCTION TEST
TEST DESCRIPTION:
    TEST IF THE DRIVE CAN PERFORM
    A WRITE CHECK FUNCTION WITHOUT
    HANGING THE DRIVE.
    A WRITE CHECK FUNCTION IS WRITTEN
    INTO MLCS1.
    THEN GO AND ERROR BITS ARE
    CHECKED FOR CORRECT STATES
!--
Local
DODU_FLG;                !DROP UNIT FLAG
BGNSUB;
CLR MBUS;
DODU_FLG = ZERO;
FIRST_BLK_XFER ();      !SET UP A FIRST BLK XFERR
MLCS1 = WRT_CHK;        !DO A WRITE CHECK FUNCTION
if .GO IS_NOT_SET       !SEE IF THE GO BIT GOT SET
then
begin
ERRDF (14, ASYNC, DUMPER); !ERROR IF NOT SET
PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_4, WRD_19);
if .DRY IS_NOT_SET      !SEE IF THE DRY IS SET WHILE GO IS CLEARED
then
begin
ERRDF (15, ASYNC, DUMPER); !ERROR IF CLEARED
PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
end;
DODU_FLG = ONE;        !SET DODU_FLG
end
else
if .DRY IS_SET          !THE GO IS SET SO SEE IF DRY IS CLEARED
then
begin
ERRDF (16, ASYNC, DUMPER); !ERROR IF DRY IS SET

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (65)

```

10722 :ML4AD
10723 :
10724 :
10725 : 5625
10726 : 5626
10727 : 5627
10728 : 5628
10729 : 5629
10730 : 5630
10731 : 5631
10732 : 5632
10733 : 5633
10734 : 5634
10735 : 5635
10736 : 5636
10737 : 5637
10738 : 5638
10739 : 5639
10740 : 5640
10741 : 5641
10742 : 5642
10743 : 5643
10744 : 5644
10745 : 5645
10746 : 5646
10747 : 5647
10748 : 5648
10749 : 5649
10750 : 5650
10751 : 5651
10752 : 5652
10753 : 5653
10754 : 5654
10755 : 5655
10756 : 5656
10757 : 5657
10758 : 5658
10759 : 5659
10760 : 5660
10761 : 5661
10762 : 5662
10763 : 5663
10764 : 5664
10765 : 5665
10766 : 5666
10767 : 5667
10768 : 5668
10769 : 5669
10770 : 5670
10771 : 5671
10772 : 5672
10773 : 5673
10774 : 5674
10775 : 5675
10776 : 5676

TEST CODE SECTION

PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
DODU_FLG = ONE;
end;

if .ILF IS_SET
then
begin
ERRDF (17, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_4, WRD_19);
end;

if .OPI IS_SET
then
begin
ERRDF (18, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_4, WRD_19);
end;

DELAY (FRTY_US);

if .DRY IS_NOT_SET
then
begin
if .GO IS_SET
then
begin
CLR_MBUS;
if .GO IS_SET then ERRDF (19, ASYNC, DUMPER) else ERRDF (20, SYNC, DUMPER);

PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19);
end
else
begin
ERRDF (21, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_4, WRD_19);
end;

DODU_FLG = ONE;

if .GO IS_SET
then
begin
CLR_MBUS;
if .GO IS_SET then ERRDF (22, ASYNC, DUMPER) else ERRDF (23, SYNC, DUMPER);

PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_4, WRD_19);
end;

```

!SEE IF ILF IS SET

!ERROR IF SET

!SEE IF OPI IS SET

!ERROR IF SET

!WAIT FO XFERR TO COMPLTE

!SEE IF DRY IS SETS AFTER XFERR

!IF DRY IS NOT SET THEN SEE IF GO IS SET

!IF THE GO IS SET THEN TRY TO CLR GO

!IF GO IS STILL SET THEN ASYNC FAILURE
!ELSE SYNC MODLE FAILURE

!ERROR GO AND DRY BOTH CLEARED

!SET THE DODU_FLG

!SEE IF THE GO IS STILL SET

!TRY TO CLR GO IF STILL SET

!IF GO IS STILL SET THE ASYNC FAILURE
!ELSE SYNC MODULE FAILURE

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (65)

```

10778 :ML4AD
10779 :
10780 : TEST CODE SECTION
10781 : 5677 DODU_FLG = ONE;
10782 : 5678 end;
10783 : 5679
10784 : 5680 ENDSLUB;
10785 : 5681
10786 : 5682 if .DODU_FLG IS_SET
10787 : 5683 then
10788 : 5684 begin
10789 : 5685 DODU (.ML_LUN);
10790 : 5686 DOCLN;
10791 : 5687 end;
10792 : 5688
10793 : 5689 ENDTST;
10797 :
10798 :

```

!DROP THIS UNIT IF DODU_FLG IS_SET

10802	037102	004167	145432	\$T13:	.SBTTL	\$T13 TEST CODE SECTION		
10803	037106	005746			JSR	R1,\$SAVE2	:	5571
10804	037110	104402		1\$:	TST	-(SP)	:	
10805	037112	152777	000040	156324	TRAP	2	:	5596
10806	037120	016701	156706		BISB	#40,@ML.REG+40	:	5598
10807	037124	042701	177770		MOV	ML,DUT,R1	:	
10808	037130	142777	000007	156306	BIC	#177770,R1	:	
10809	037136	150177	156302		BICB	#7,@ML.REG+40	:	
10810	037142	005002			BISB	R1,@ML.REG+40	:	
10811	037144	004767	161112		CLR	R2	:	DODU.FLG
10812	037150	012777	000051	156226	JSR	PC,FIRST.BLK.XFER	:	5600
10813	037156	132777	000001	156220	MOV	#51,@ML.REG	:	5601
10814	037164	001057			BITB	#1,@ML.REG	:	5602
10815	037166	104455			BNE	2\$:	5604
10816	037170	000016			TRAP	55	:	
10817	037172	012706			.WORD	16	:	5607
10818	037174	026302			.WORD	ASYN	:	
10819	037176	012746	010730		.WORD	DUMPER	:	
10820	037202	012746	012200		MOV	#WRD.19,-(SP)	:	
10821	037206	012746	010650		MOV	#FNC.4,-(SP)	:	5608
10822	037212	012746	011610		MOV	#WRD.12,-(SP)	:	
10823	037216	012746	010520		MOV	#PHR.1,-(SP)	:	
10824	037222	012746	010304		MOV	#WRD.1,-(SP)	:	
10825	037226	012746	000006		MOV	#FIV.FMT,-(SP)	:	
10826	037232	010600			MOV	#6,-(SP)	:	
10827	037234	104414			MOV	SP,R0	:	SP,*
10828	037236	132777	000200	156210	TRAP	14	:	
10829	037244	001056			BITB	#200,@ML.REG+50	:	5610
10830	037246	104455			BNE	3\$:	
10831	037250	000017			TRAP	55	:	5613
10832	037252	012706			.WORD	17	:	
					.WORD	ASYN	:	

Address	Hex	Hex	Hex	Label	Instruction	Comments	Seq
10946							
10947							
10948							
10949	037764	012746	010730		MOV #WRD.19,-(SP)		
10950	037770	012746	012200		MOV #FNC.4,-(SP)		
10951	037774	012746	010640		MOV #WRD.11,-(SP)		
10952	040000	012746	011610		MOV #PHR.1,-(SP)		
10953	040004	012746	010524		MOV #WRD.2,-(SP)		
10954	040010	012746	010304		MOV #FIV.FMT,-(SP)		
10955	040014	012746	000006		MOV #6,-(SP)		
10956	040020	010600			MOV SP,R0		
10957	040022	104414			TRAP 14	: SP,*	
10958	040024	012702	000001	14\$:	MOV #1,R2	: *.DODU.FLG	
10959	040030	062706	000016		ADD #16,SP	:	5665
10960	040034	132777	000001	155342 15\$:	BITB #1,@ML.REG	:	5647
10961	040042	001455			BEQ 18\$:	5668
10962	040044	152777	000040	155372	BISB #40,@ML.REG+40	:	
10963	040052	016701	155754		MOV ML.DUT,R1		5670
10964	040056	042701	177770		BIC #177770,R1		
10965	040062	142777	000007	155354	BICB #7,@ML.REG+40		
10966	040070	150177	155350		BISB R1,@ML.REG+40		
10967	040074	132777	000001	155302	BITB #1,@ML.REG		
10968	040102	001405			BEQ 16\$:	5673
10969	040104	104455			TRAP 55		
10970	040106	000026			.WORD 26		
10971	040110	012706			.WORD ASYNC		
10972	040112	026302			.WORD DUMPER		
10973	040114	000404			.WORD		
10974	040116	104455		16\$:	ER 17\$		
10975	040120	000027			TRAP 55		
10976	040122	012750			.WORD 27		
10977	040124	026302			.WORD SYNC		
10978	040126	012746	010730	17\$:	.WORD DUMPER		
10979	040132	012746	012200		MOV #WRD.19,-(SP)	:	5676
10980	040136	012746	010640		MOV #FNC.4,-(SP)		
10981	040142	012746	011626		MOV #WRD.11,-(SP)		
10982	040146	012746	010520		MOV #PHR.2,-(SP)		
10983	040152	012746	010304		MOV #WRD.1,-(SP)		
10984	040156	012746	000006		MOV #FIV.FMT,-(SP)		
10985	040162	010600			MOV #6,-(SP)		
10986	040164	104414			MOV SP,R0	: SP,*	
10987	040166	012702	000001		TRAP 14		
10988	040172	062706	000016		MOV #1,R2	: *.DODU.FLG	5677
10989	040176	104467		18\$:	ADD #16,SP	:	5670
10990	040200	006000			TRAP 67	:	5678
10991	040202	103002			ROR R0		
10992	040204	000167	176700		BHIS 19\$		
10993	040210	005302		19\$:	JMP 1\$		
10994	040212	001004			DEC R2	: DODU.FLG	5682
10995	040214	016700	155610		BNE 20\$		
10996	040220	104451			MOV ML.LUN,R0	:	5685
10997	040222	104444			TRAP 51		
10998	040224	005726		20\$:	TRAP 44		
10999	040226	000207			TST (SP)+	:	5571
10000					RTS PC		

```
11002      :ML4AD
11003      :
11004      : TEST CODE SECTION
11005      : Routine Size: 299 words
11006      : Maximum stack depth per invocation: 18 words
11011
11012
11016
11017
11021 040230      :
11022 040230 004767 176646      :
11023 040234 104466      :
11024 040236 006000      :
11025 040240 103773      :
11026 040242 000207      :
11027
11028      :
11029      : Routine Size: 6 words
11034      : Maximum stack depth per invocation: 0 words
11035
11036 :          5690 !<BLF/PAGE>
```

T13:: .SBTTL T13 TEST CODE SECTION

```
T13:: JSR PC,ST13
1$: TRAP 66
ROR R0
BLO 1$
RTS PC
```

5687

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (66)

```

11038 :ML4AD
11039 :
11040 :
11041 : 5691 !
11042 : 5692 !BGNTST;
11043 : 5693 !
11044 : 5694 !++
11045 : 5695 !TEST NUMBER: TST 14
11046 : 5696 !
11047 : 5697 !TEST NAME: WRITE FUNCTION TEST
11048 : 5698 !
11049 : 5699 !TEST DESCRIPTION:
11050 : 5700 !TEST IF THE DRIVE CAN PERFORM A WRITE FUNCTION WITHOUT
11051 : 5701 !HANGING THE DRIVE.
11052 : 5702 !
11053 : 5703 !A WRITE FUNCTION IS WRITTEN INTO MLCS1. THEN GO AND ERROR BITS ARE
11054 : 5704 !CHECKED FOR CORRECT STATUS. THIS UNIT IS DROPPED ON DETECTED ERRORS.
11055 : 5705 !--
11056 : 5706 !
11057 : 5707 !local
11058 : 5708 !DODU_FLG; !DROP UNIT FLAG
11059 : 5709 !
11060 : 5710 !BGNSUB;
11061 : 5711 !CLR MBUS;
11062 : 5712 !DODU_FLG = ZERO;
11063 : 5713 !FIRST_BLK_XFER (); !SET UP A FIRST BLOCK XFERR
11064 : 5714 !MLCS1 = write; !DO A WRITE FUNCTION
11065 : 5715 !
11066 : 5716 !if .GO IS_NOT_SET !SEE IF THE GO IS SET
11067 : 5717 !then !
11068 : 5718 !begin !ERROR IF NOT SET
11069 : 5719 !ERRDF (24, ASYNC, DUMPER);
11070 : 5720 !PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_5, WRD_19);
11071 : 5721 !
11072 : 5722 !if .DRY IS_NOT_SET !SEE IF DRY SET WITH GO CLEAR
11073 : 5723 !then !
11074 : 5724 !begin !
11075 : 5725 !ERRDF (25, ASYNC, DUMPER);
11076 : 5726 !PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
11077 : 5727 !end; !
11078 : 5728 !
11079 : 5729 !DODU_FLG = ONE; !SET DODU_FLG
11080 : 5730 !end !
11081 : 5731 !else !GO IS SET DURING FUNCTION
11082 : 5732 !
11083 : 5733 !if .DRY IS_SET !SEE IF DRY CLEAR WITH GO SET
11084 : 5734 !then !
11085 : 5735 !begin !ERROR IF SET
11086 : 5736 !ERRDF (26, ASYNC, DUMPER);
11087 : 5737 !PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_5);
11088 : 5738 !DODU_FLG = ONE;
11089 : 5739 !end; !
11090 : 5740 !
11091 : 5741 !if .ILF IS_SET !DID FUNCTION CAUSE ILF
11092 : 5742 !then !
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (66)

```

11094 :ML4AD
11095 :
11096 :
11097 : 5743 begin
11098 : 5744 ERRDF (27, ASYNC, DUMPER);
11099 : 5745 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_5, WRD_19);
11100 : 5746 end;
11101 : 5747
11102 : 5748 if .OPI IS_SET
11103 : 5749 then
11104 : 5750 begin
11105 : 5751 ERRDF (28, ASYNC, DUMPER);
11106 : 5752 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_5, WRD_19);
11107 : 5753 end;
11108 : 5754
11109 : 5755 DELAY (FRTY_US);
11110 : 5756
11111 : 5757 if .DRY IS_NOT_SET
11112 : 5758 then
11113 : 5759 begin
11114 : 5760
11115 : 5761 if .GO IS_SET
11116 : 5762 then
11117 : 5763 begin
11118 : 5764 CLR_MBUS;
11119 : 5765
11120 : 5766 if .GO IS_SET then ERRDF (29, ASYNC, DUMPER) else ERRDF (30, SYNC, DUMPER);
11121 : 5767
11122 : 5768 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
11123 : 5769 end
11124 : 5770
11125 : 5771 else
11126 : 5772 begin
11127 : 5773 ERRDF (31, ASYNC, DUMPER);
11128 : 5774 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_5, WRD_19);
11129 : 5775 end;
11130 : 5776
11131 : 5777 DODU_FLG = ONE;
11132 : 5778 end;
11133 : 5779
11134 : 5780 if .GO IS_SET
11135 : 5781 then
11136 : 5782 begin
11137 : 5783 CLR_MBUS;
11138 : 5784
11139 : 5785 if .GO IS_SET then ERRDF (32, ASYNC, DUMPER) else ERRDF (33, SYNC, DUMPER);
11140 : 5786
11141 : 5787 PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_5, WRD_19);
11142 : 5788 DODU_FLG = ONE;
11143 : 5789 end;
11144 : 5790
11145 : 5791 ENDSUB;
11146 : 5792
11147 : 5793
11148 : 5794 if .TRE IS_SET
  
```

```

!ERROR IF YES
!DID FUNCTION CAUSE OPI
!ERROR IF YES
!WAIT FOR XFERR TO COMPLETE
!SEE IF DRY CLEARED AFTER XFERR
!TST GO CLR IF DRY NOT SET
!CLEAR GO IF STILL SET
!TST GO TO DETERMINE FAILING MOD
!DRY NOT SET AND GO CLEARED
!REPORT ERROR
!SET DODU_FLG
!SEE IF GO CLEARED AFTER XFERR.
!CLEAR GO IF STILL SET
!TST GO TO DETERMINE FAILING MOD
!SEE IF XFERR CAUSED A TRANSFER ERROR
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (66)

```

11150 :ML4AD
11151 :
11152 :
11153 : 5795 then
11154 : 5796 begin
11155 : 5797 ERRDF (115, INTER, DUMPER);
11156 : 5798 PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_5, WRD_19);
11157 : 5799 end;
11158 : 5800
11159 : 5801 if .DODU_FLG IS_SET
11160 : 5802 then
11161 : 5803 begin
11162 : 5804 DODU (.ML_LUN);
11163 : 5805 DOCLN;
11164 : 5806 end;
11165 : 5807
11166 : 5808 ENDTST;
11170 :
11171 :

```

!REPORT ERROR IF SET AND CONTINUE TESTING

!DROP THIS UNIT IF DODU_FLG SET

Address	Hex	Dec	Label	Instruction	Comments	Address
11175	040244	004167	144270	\$T14: JSR	R1,\$SAVE2	
11176	040250	005746		TST	-(SP)	5689
11177	040252	104402		1\$: TRAP	2	
11178	040254	152777	000040 155162	BISB	#40,@ML.REG+40	5708
11179	040262	016701	155544	MOV	ML.DUT,R1	5710
11180	040266	042701	177770	BIC	#177770,R1	
11181	040272	142777	000007 155144	BICB	#7,@ML.REG+40	
11182	040300	150177	155140	BISB	R1,@ML.REG+40	
11183	040304	005002		CLR	R2	: DODU.FLG
11184	040306	004767	157750	JSR	PC,FIRST.BLK.XFER	5712
11185	040312	012777	000061 155064	MOV	#61,@ML.REG	5713
11186	040320	132777	000001 155056	BITB	#1,@ML.REG	5714
11187	040326	001057		BNE	2\$	5716
11188	040330	104455		TRAP	55	
11189	040332	000030		.WORD	30	: 5719
11190	040334	012706		.WORD	ASYN	
11191	040336	026302		.WORD	DUMPER	
11192	040340	012746	010730	MOV	#WRD.19,-(SP)	
11193	040344	012746	012216	MOV	#FNC.5,-(SP)	: 5720
11194	040350	012746	010650	MOV	#WRD.12,-(SP)	
11195	040354	012746	011610	MOV	#PHR.1,-(SP)	
11196	040360	012746	010520	MOV	#WRD.1,-(SP)	
11197	040364	012746	010304	MOV	#FIV.FMT,-(SP)	
11198	040370	012746	000006	MOV	#6,-(SP)	
11199	040374	010600		MOV	SP,R0	: SP,*
11200	040376	104414		TRAP	14	
11201	040400	132777	000200 155046	BITB	#200,@ML.REG+50	
11202	040406	001056		BNE	3\$: 5722
11203	040410	104455		TRAP	55	
11204	040412	000031		.WORD	31	: 5725

Address	OpCode	Operand1	Operand2	Label	Instruction	Comments	Page
11206							
11207							
11208							
11209	040414	012706			.WORD ASYNC		
11210	040416	026302			.WORD DUMPER		
11211	040420	012746	011726		MOV #PHR.6,-(SP)		
11212	040424	012746	010520		MOV #WRD.1,-(SP)		5726
11213	040430	012746	011176		MOV #WRD.43,-(SP)		
11214	040434	012746	011610		MOV #PHR.1,-(SP)		
11215	040440	012746	010524		MOV #WRD.2,-(SP)		
11216	040444	012746	010304		MOV #FIV.FMT,-(SP)		
11217	040450	012746	000006		MOV #6,-(SP)		
11218	040454	010600			MOV SP,R0	: SP,*	
11219	040456	104414			TRAP 14		
11220	040460	062706	000016		ADD #16,SP		
11221	040464	000427			BR 3\$		5724
11222	040466	105777	154762	2\$:	TSTB @ML.REG+50		5729
11223	040472	100030			BPL 4\$		5733
11224	040474	104455			TRAP 55		
11225	040476	000032			.WORD 32		5736
11226	040500	012706			.WORD ASYNC		
11227	040502	026302			.WORD DUMPER		
11228	040504	012746	011714		MOV #PHR.5,-(SP)		
11229	040510	012746	010520		MOV #WRD.1,-(SP)		5737
11230	040514	012746	011176		MOV #WRD.43,-(SP)		
11231	040520	012746	011626		MOV #PHR.2,-(SP)		
11232	040524	012746	010524		MOV #WRD.2,-(SP)		
11233	040530	012746	010304		MOV #FIV.FMT,-(SP)		
11234	040534	012746	000006		MOV #6,-(SP)		
11235	040540	010600			MOV SP,R0	: SP,*	
11236	040542	104414			TRAP 14		
11237	040544	012702	000001	3\$:	MOV #1,R2	: *,DODU.FLG	5738
11238	040550	062706	000016		ADD #16,SP		5735
11239	040554	132777	000001	154702 4\$:	BITB #1,@ML.REG+60		5741
11240	040562	001426			BEQ 5\$		
11241	040564	104455			TRAP 55		
11242	040566	000033			.WORD 33		5744
11243	040570	012706			.WORD ASYNC		
11244	040572	026302			.WORD DUMPER		
11245	040574	012746	010730		MOV #WRD.19,-(SP)		
11246	040600	012746	012216		MOV #FNC.5,-(SP)		5745
11247	040604	012746	010650		MOV #WRD.12,-(SP)		
11248	040610	012746	011714		MOV #PHR.5,-(SP)		
11249	040614	012746	010536		MOV #WRD.3,-(SP)		
11250	040620	012746	010304		MOV #FIV.FMT,-(SP)		
11251	040624	012746	000006		MOV #6,-(SP)		
11252	040630	010600			MOV SP,R0	: SP,*	
11253	040632	104414			TRAP 14		
11254	040634	062706	000016		ADD #16,SP		
11255	040640	032777	020000	154616 5\$:	BIT #20000,@ML.REG+60		5743
11256	040646	001426			BEQ 6\$		5748
11257	040650	104455			TRAP 55		
11258	040652	000034			.WORD 34		5751
11259	040654	012706			.WORD ASYNC		
11260	040656	026302			.WORD DUMPER		

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA.<

11374
11375
11376
11377 041376 012746 012216
11378 041402 012746 010650
11379 041406 012746 011714
11380 041412 012746 010736
11381 041416 012746 011416
11382 041422 012746 010322
11383 041426 012746 000007
11384 041432 010600
11385 041434 104414
11386 041'36 062706 000020
11387 0414+2 005302
11388 041444 001004
11389 041446 016700 154356
11390 041452 104451
11391 041454 104444
11392 041456 005726
11393 041460 000207
11394
11395
11396
11401
11402
11406
11407
11411 041462
11412 041462 004767 176556
11413 041466 104466
11414 041470 006000
11415 041472 103773
11416 041474 000207
11417
11418
11419
11424
11425
11426 :

:ML4AD
:

TEST CODE SECTION

```

MOV #FNC.5,-(SP)
MOV #WRD.12,-(SP)
MOV #PHR.5,-(SP)
MOV #WRD.20,-(SP)
MOV #WRD.61,-(SP)
MOV #SIX.FMT,-(SP)
MOV #7,-(SP)
MOV SP,R0
TRAP 14
ADD #20,SP
20$: DEC R2
BNE 21$
MOV ML.LUN,R0
TRAP 51
TRAP 44
21$: TST (SP)+
RTS PC

```

: SP,*
:
: DODU.FLG
:
:
:

5796
5801
5804
5689

: Routine Size: 327 words
: Maximum stack depth per invocation: 18 words

.SBTTL T14 TEST CODE SECTION

```

T14::
1$: JSR PC,$T14
TRAP 66
ROR R0
BLO 1$
RTS PC

```

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

5806

5809 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (67)

```

11428 :ML4AD
11429 :
11430 : TEST CODE SECTION
11431 : 5810 :
11432 : 5811 :
11433 : 5812 BGNTST;
11434 : 5813 :
11435 : 5814 !++
11436 : 5815 TEST NUMBER: TST 15
11437 : 5816 :
11438 : 5817 TEST NAME: READ FUNCTION TEST
11439 : 5818 :
11440 : 5819 TEST DESCRIPTION:
11441 : 5820 :
11442 : 5821 TEST IF THE DRIVE CAN PERFORM
11443 : 5822 A READ FUNCTION WITHOUT
11444 : 5823 HANGING THE DRIVE.
11445 : 5824 :
11446 : 5825 A READ FUNCTION IS WRITTEN
11447 : 5826 INTO MLCS1
11448 : 5827 :
11449 : 5828 THEN GO AND ERROR BITS ARE
11450 : 5829 CHECKED FOR CORRECT STATES.
11451 : 5830 :
11452 : 5831 !--
11453 : 5832 :
11454 : 5833 local
11455 : 5834 DODU_FLG; !DROP UNIT FLAG
11456 : 5835 :
11457 : 5836 BGNSUB;
11458 : 5837 CLR MBUS;
11459 : 5838 DODU_FLG = ZERO;
11460 : 5839 FIRST_BLK_XFER (); !SET UP A FIRST BLK XFERR
11461 : 5840 MLCS1 = read; !DO A READ FUNCTION
11462 : 5841 :
11463 : 5842 if .GO IS_NOT_SET !SEE IF GO GOT SET
11464 : 5843 then
11465 : 5844 begin !ERROR IF CLEAR
11466 : 5845 ERRDF (34, ASYNC, DUMPER);
11467 : 5846 PRINTB (FIV_FMT, WRD_1, PHR_1, WRD_12, FNC_6, WRD_19);
11468 : 5847 :
11469 : 5848 if .DRY IS_NOT_SET !TST DRY SET WITH GO CLEAR
11470 : 5849 then
11471 : 5850 begin !ERROR IF NOT SET
11472 : 5851 ERRDF (35, ASYNC, DUMPER);
11473 : 5852 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
11474 : 5853 end;
11475 : 5854 :
11476 : 5855 DODU_FLG = One; !SET DODU_FLG
11477 : 5856 end
11478 : 5857 else !GO BIT GOT SET
11479 : 5858 :
11480 : 5859 if .DRY IS_SET !SEE IF DRY IS CLEAR
11481 : 5860 then
11482 : 5861 begin !ERROR IF SET

```



```

11484 :ML4AD
11485 :
11486 :
11487 : 5862      ERRDF (36, ASYNC, DUMPER);
11488 : 5863      PRINTB (FIV_FMT, WRD_2, PHR_2, WRD_43, WRD_1, PHR_6);
11489 : 5864      DODU_FLG = ONE;
11490 : 5865      end;
11491 : 5866
11492 : 5867      if .ILF IS_SET                !DID FUNCTION CAUSE ILF
11493 : 5868      then
11494 : 5869      begin                          !ERROR IF YES
11495 : 5870      ERRDF (37, ASYNC, DUMPER);
11496 : 5871      PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_6, WRD_19);
11497 : 5872      end;
11498 : 5873
11499 : 5874      if .OPI IS_SET                !DID FUNCTION CAUSE OPI
11500 : 5875      then
11501 : 5876      begin                          !ERROR IF YES
11502 : 5877      ERRDF (38, ASYNC, DUMPER);
11503 : 5878      PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_6, WRD_19);
11504 : 5879      end;
11505 : 5880
11506 : 5881      DELAY (FRTY_US);                !WAIT FOR XFERR TO COMPLETE
11507 : 5882
11508 : 5883      if .DRY IS_NOT_SET              !IS DRY SET AFTER XFERR
11509 : 5884      then
11510 : 5885      begin
11511 : 5886
11512 : 5887      if .GO IS_SET                    !TEST GO CLEAR WITH DRY NOT SET
11513 : 5888      then
11514 : 5889      begin
11515 : 5890      CLR_MBUS;                       !CLEAR GO
11516 : 5891
11517 : 5892      if .GO IS_SET then ERRDF (39, ASYNC, DUMPER) else ERRDF (40, SYNC, DUMPER);
11518 : 5893
11519 : 5894
11520 : 5895      PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_6, WRD_19);
11521 : 5896      end
11522 : 5897      else
11523 : 5898      begin
11524 : 5899      ERRDF (41, ASYNC, DUMPER);        !GO BIT CLEAR WITH DRY NOT SET
11525 : 5900      PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_11, FNC_6, WRD_19);    !REPORT ERROR
11526 : 5901      end;
11527 : 5902
11528 : 5903      DODU_FLG = ONE;                  !SET DODU_FLG
11529 : 5904      end;
11530 : 5905
11531 : 5906      if .GO IS_SET                    !SEE IF GO CLEAR AFTER XFERR
11532 : 5907      then
11533 : 5908      begin
11534 : 5909      CLR_MBUS;                       !CLEAR GO IF STILL SET
11535 : 5910
11536 : 5911      if .GO IS_SET then ERRDF (42, ASYNC, DUMPER) else ERRDF (43, SYNC, DUMPER);
11537 : 5912
11538 : 5913
11538 :
  
```

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (67)

11540 :ML4AD

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (67)

```

11541 :
11542 :
11543 :      5914      PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_6, WRD_19);
11544 :      5915      DODU_FLG = ONE;
11545 :      5916      end;
11546 :      5917
11547 :      5918      ENDSUB;
11548 :      5919
11549 :      5920      if .TRE IS_SET
11550 :      5921      then
11551 :      5922          begin
11552 :      5923              ERRDF (116, INTER, DUMPER);
11553 :      5924              PRINTB (SIX_FMT, WRD_61, WRD_20, PHR_5, WRD_12, FNC_6, WRD_19);
11554 :      5925          end;
11555 :      5926
11556 :      5927      if .DODU_FLG IS_SET
11557 :      5928      then
11558 :      5929          begin
11559 :      5930              DODU (.ML_LUN);
11560 :      5931              DOCLN;
11561 :      5932          end;
11562 :      5933
11563 :      5934      ENDTST;

```

!SEE IF XFERR CAUSED A TRANSFER ERROR
!REPORT ERROR IF SET AND CONTINUE TESTING
!DROP THIS UNIT IF DODU_FLG SET

Address	Offset	Hex	Dec	Label	Instruction	Comment	Address
11572	041476	004167	143036	ST15:	.SBTTL	\$T15 TEST CODE SECTION	
11573	041502	005746			JSR	R1,\$SAVE2	5808
11574	041504	104402		1\$:	TST	-(SP)	
11575	041506	152777	000040 153730		TRAP	2	5834
11576	041514	016701	154312		BISB	#40,@ML.REG+40	5836
11577	041520	042701	177770		MOV	ML.DUT,R1	
11578	041524	142777	000007 153712		BIC	#177770,R1	
11579	041532	150177	153706		BICB	#7,@ML.REG+40	
11580	041536	0C5002			BISB	R1,@ML.REG+40	
11581	041540	004767	156516		CLR	R2	: DODU.FLG 5838
11582	041544	012777	000071 153632		JSR	PC,FIRST.BLK.XFER	5839
11583	041552	132777	000001 153624		MOV	#71,@ML.REG	5840
11584	041560	001057			BITB	#1,@ML.REG	5842
11585	041562	104455			BNE	2\$	
11586	041564	000042			TRAP	55	5845
11587	041566	012706			.WORD	42	
11588	041570	026302			.WORD	ASYN	
11589	041572	012746	010730		.WORD	DUMPER	
11590	041576	012746	012226		MOV	#WRD.19,-(SP)	5846
11591	041602	012746	010650		MOV	#FNC.6,-(SP)	
11592	041606	012746	011610		MOV	#WRD.12,-(SP)	
11593	041612	012746	010520		MOV	#PHR.1,-(SP)	
11594	041616	012746	010304		MOV	#WRD.1,-(SP)	
					MOV	#FIV.FMT,-(SP)	

Address	OpCode	Op1	Op2	Op3	Op4	Instruction	Comments	Line
11708								
11709								
11710								
11711	042336	012746	000006			MOV #6,-(SP)		
11712	042342	010600				MOV SP,R0		
11713	042344	104414				TRAP 14	: SP,*	
11714	042346	000424				BR 14\$:	
11715	042350	104455				TRAP 55	:	5887
11716	042352	000051				.WORD 51	:	5899
11717	042354	012706				.WORD ASYNC		
11718	042356	026302				.WORD DUMPER		
11719	042360	012746	010730			MOV #WORD.19,-(SP)	:	
11720	042364	012746	012226			MOV #FNC.6,-(SP)	:	5900
11721	042370	012746	010640			MOV #WORD.11,-(SP)		
11722	042374	012746	011610			MOV #PHR.1,-(SP)		
11723	042400	012746	010524			MOV #WORD.2,-(SP)		
11724	042404	012746	010304			MOV #FIV.FMT,-(SP)		
11725	042410	012746	000006			MOV #6,-(SP)		
11726	042414	010600				MOV SP,R0	: SP,*	
11727	042416	104414				TRAP 14	:	
11728	042420	012702	000001			MOV #1,R2	: *.DODU.FLG	
11729	042424	062706	000016			ADD #16,SP	:	5903
11730	042430	132777	000001	152746		BITB #1,@ML.REG	:	5885
11731	042436	001455				BEQ 18\$:	5906
11732	042440	152777	000040	152776		BISB #40,@ML.REG+40	:	
11733	042446	016701	153360			MOV ML.DUT,R1	:	5908
11734	042452	042701	177770			BIC #177770,R1		
11735	042456	142777	000007	152760		BICB #7,@ML.REG+40		
11736	042464	150177	152754			BISB R1,@ML.REG+40		
11737	042470	132777	000001	152706		BITB #1,@ML.REG	:	
11738	042476	001405				BEQ 16\$:	5911
11739	042500	104455				TRAP 55		
11740	042502	000052				.WORD 52		
11741	042504	012706				.WORD ASYNC		
11742	042506	026302				.WORD DUMPER		
11743	042510	000404				BR 17\$		
11744	042512	104455				TRAP 55	:	
11745	042514	000053				.WORD 53		
11746	042516	012750				.WORD SYNC		
11747	042520	026302				.WORD DUMPER		
11748	042522	012746	010730			MOV #WORD.19,-(SP)	:	
11749	042526	012746	012226			MOV #FNC.6,-(SP)	:	5914
11750	042532	012746	010640			MOV #WORD.11,-(SP)		
11751	042536	012746	011626			MOV #PHR.2,-(SP)		
11752	042542	012746	010520			MOV #WORD.1,-(SP)		
11753	042546	012746	010304			MOV #FIV.FMT,-(SP)		
11754	042552	012746	000006			MOV #6,-(SP)		
11755	042556	010600				MOV SP,R0	: SP,*	
11756	042560	104414				TRAP 14	:	
11757	042562	012702	000001			MOV #1,R2	: *.DODU.FLG	
11758	042566	062706	000016			ADD #16,SP	:	5915
11759	042572	104467				TRAP 67	:	5908
11760	042574	006000				ROR R0	:	5916
11761	042576	103002				BHIS 19\$		
11762	042600	000167	176700			JMP 1\$		

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

```

11764      ;ML4AD
11765      :
11766      TEST CODE SECTION
11767 042604 032777 040000 152572 19$: BIT #40000,ML.REG ;
11768 042612 001430 BEQ 20$ ; 5920
11769 042614 104455 TRAP 55 ;
11770 042616 000164 .WORD 164 ; 5923
11771 042620 013114 .WORD INTER
11772 042622 026302 .WORD DUMPER
11773 042624 012746 010730 MOV #WRD.19,-(SP) ;
11774 042630 012746 012226 MOV #FNC.6,-(SP) ; 5924
11775 042634 012746 010650 MOV #WRD.12,-(SP)
11776 042640 012746 011714 MOV #PHR.5,-(SP)
11777 042644 012746 010736 MOV #WRD.20,-(SP)
11778 042650 012746 011416 MOV #WRD.61,-(SP)
11779 042654 012746 010322 MOV #SIX.FMT,-(SP)
11780 042660 012746 000007 MOV #7,-(SP)
11781 042664 010600 MOV SP,R0 ; SP,*
11782 042666 104414 TRAP 14 ;
11783 042670 062706 000020 ADD #20,SP ;
11784 042674 005302 20$: DEC R2 ; 5922
11785 042676 001004 BNE 21$ ; DODU.FLG 5927
11786 042700 016700 153124 MOV ML,LUN,R0 ;
11787 042704 104451 TRAP 51 ; 5930
11788 042706 104444 TRAP 44 ;
11789 042710 005726 21$: TST (SP)+ ;
11790 042712 000207 RTS PC ; 5808

```

: Routine Size: 327 words
: Maximum stack depth per invocation: 18 words

```

11791
11792
11793
11798
11799
11803
11804      .SBTTL T15 TEST CODE SECTION
11808 042714      T15::
11809 042714 004767 176556 1$: JSR PC,$T15 ;
11810 042720 104466 TRAP 66 ; 5932
11811 042722 006000 ROR R0
11812 042724 103773 BLO 1$
11813 042726 000207 RTS PC

```

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

11814
11815
11816

11825
11826
11827 ; 5935 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (68)

```

11829 :ML4AD
11830 :
11831 :
11832 : 5936 !
11833 : 5937 TEST CODE SECTION
11834 : 5938 BGNTS ;
11835 : 5939 !++
11836 : 5940 TEST NUMBER: TST 16
11837 : 5941
11838 : 5942 TEST NAME: CLEAR FUNCTION TEST
11839 : 5943
11840 : 5944 TEST DESCRIPTION:
11841 : 5945 TEST IF THE DRIVE CAN PERFORM A CLEAR FUNCTION WITHOUT HANGING THE DRIVE.
11842 : 5946
11843 : 5947 A CLEAR FUNCTION IS WRITTEN INTO MLCS1.
11844 : 5948
11845 : 5949 THEN GO AND ERROR BITS ARE CHECKED FOR CORRECT STATUS.
11846 : 5950 THIS DRIVE IS DROPPED ON DETECTED ERRORS.
11847 : 5951
11848 : 5952 !--
11849 : 5953
11850 : 5954 BGNSUB;
11851 : 5955 CLR MBUS;
11852 : 5956 MLER = ONES;
11853 : 5957 MLCS1 = DRV CLR;
11854 : 5958 DELAY (ONE_US);
11855 : 5959
11856 : 5960 if .GO IS_SET
11857 : 5961 then
11858 : 5962 begin
11859 : 5963 ERRDF (44, ASYNC, DUMPER);
11860 : 5964 PRINTB (FIV_FMT, WRD_1, PHR_5, WRD_11, FNC_3, FNC_7, WRD_19);
11861 : 5965
11862 : 5966 if .DRY IS_SET
11863 : 5967 then
11864 : 5968 begin
11865 : 5969 ERRDF (45, ASYNC, DUMPER);
11866 : 5970 PRINTB (FIV_FMT, WRD_2, PHR_5, WRD_43, WRD_1, PHR_5);
11867 : 5971 end;
11868 : 5972
11869 : 5973 end
11870 : 5974 else
11871 : 5975
11872 : 5976 if .DRY IS_NOT_SET
11873 : 5977 then
11874 : 5978 begin
11875 : 5979 ERRDF (46, ASYNC, DUMPER);
11876 : 5980 PRINTB (FIV_FMT, WRD_2, PHR_1, WRD_43, WRD_1, PHR_6);
11877 : 5981 end;
11878 : 5982
11879 : 5983 if .ILF IS_SET
11880 : 5984 then
11881 : 5985 begin
11882 : 5986 ERRDF (47, ASYNC, DUMPER);
11883 : 5987 PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);

```

```

!SET BITS IN ERROR REGISTER
!DO A CLEAR FUNCTION
!DELAY
!SEE IF GO CLEARED AFTER FUNCTION
!ERROR IF SET
!TST DRY CLEAR WITH GO SET
!ERROR IF SET
!GO CLEARED AFTER FUNCTION
!TST DRY SET WITH GO CLEAR
!ERROR IF NOT SET
!DID FUNCTION CAUSE ILF
!ERROR IF YES

```


29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (68)

```

11885 :ML4AD
11886 :
11887 : TEST CODE SECTION
11888 : 5988 end;
11889 : 5989
11890 : 5990 if .OPI IS_SET !DID FUNCTION CAUSE OPI
11891 : 5991 then !ERROR IF YES
11892 : 5992 begin !ERROR IF YES
11893 : 5993 ERRDF (48, ASYNC, DUMPER);
11894 : 5994 PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_3, FNC_7, WRD_19);
11895 : 5995 end;
11896 : 5996
11897 : 5997 if .MLER neq ZERO !TEST ERROR REGISTER FOR CLEAR
11898 : 5998 then !ERROR IF NOT CLEAR
11899 : 5999 begin !ERROR IF NOT CLEAR
11900 : 6000 ERRDF (49, ASYNC, DUMPER);
11901 : 6001 PRINTB (SIX_FMT, FNC_3, FNC_7, WRD_19, WRD_14, WRD_13, REG_3);
11902 : 6002 end;
11903 : 6003
11904 : 6004 ENDSUB;
11905 : 6005 ENDTST;

```

```

11910
11914 042730 010146 $T16: .SBTTL $T16 TEST CODE SECTION
11915 042732 005746 MOV R1, -(SP)
11916 042734 104402 TST 1
11917 042736 152777 000040 152500 1$: TRAP 2
11918 042744 016701 153062 BISB #40, @ML.REG+40
11919 042750 042701 177770 MOV ML, DUT, R1
11920 042754 142777 000007 152462 BIC #177770, R1
11921 042762 150177 152456 BICB #7, @ML.REG+40
11922 042766 012777 177777 152470 BISB R1, @ML.REG+40
11923 042774 012777 000011 152402 MOV #-1, @ML.REG+60
11924 043002 012700 000001 MOV #11, @ML.REG
11925 043006 001410 2$: MOV #1, R0
11926 043010 016701 137102 BEQ 5$
11927 043014 001403 MOV LSDLY, R1
11928 043016 005016 BEQ 4$
11929 043020 005301 3$: CLR (SP)
11930 043022 001375 DEC R1
11931 043024 005300 BNE 3$
11932 043026 000767 DEC R0
11933 043030 132777 000001 152346 5$: BR 2$
11934 043036 001462 BITB #1, @ML.REG
11935 043040 104455 BEQ 7$
11936 043042 000054 TRAP 55
11937 043044 012706 .WORD 54
11938 043046 026302 .WORD ASYNC
11939 043050 012746 010730 .WORD DUMPER
MOV #WRD.19, -(SP)

```

5934

5937
5954

5956
5957
5958

5960

5963

5964

Address	Hex	Hex	Hex	Label	Code	Comments	Address
11997				:ML4AD			
11998				:			
11999					TEST CODE SECTION		
12000	043324	012746	010650		MOV #WRD.12,-(SP)		
12001	043330	012746	011714		MOV #PHR.5,-(SP)		
12002	043334	012746	010536		MOV #WRD.3,-(SP)		
12003	043340	012746	010304		MOV #FIV.FMT,-(SP)		
12004	043344	012746	000007		MOV #7,-(SP)		
12005	043350	010600			MOV SP,R0		
12006	043352	104414			TRAP 14	: SP,*	
12007	043354	062706	000020		ADD #20,SP	:	
12008	043360	032777	020000	152076 9\$:	BIT #20000,@ML.REG+60	:	5985
12009	043366	001430			BEG 10\$:	5990
12010	043370	104455			TRAP 55	:	
12011	043372	000060			.WORD 60	:	5993
12012	043374	012706			.WORD ASYNC		
12013	043376	026302			.WORD DUMPER		
12014	043400	012746	010730		MOV #WRD.19,-(SP)	:	
12015	043404	012746	012234		MOV #FNC.7,-(SP)	:	5994
12016	043410	012746	012172		MOV #FNC.3,-(SP)		
12017	043414	012746	010650		MOV #WRD.12,-(SP)		
12018	043420	012746	011714		MOV #PHR.5,-(SP)		
12019	043424	012746	010544		MOV #WRD.4,-(SP)		
12020	043430	012746	010304		MOV #FIV.FMT,-(SP)		
12021	043434	012746	000007		MOV #7,-(SP)		
12022	043440	010600			MOV SP,R0	: SP,*	
12023	043442	104414			TRAP 14	:	
12024	043444	062706	000020		ADD #20,SP	:	5992
12025	043450	005777	152010	10\$:	TST @ML.REG+60	:	5997
12026	043454	001430			BEG 11\$:	
12027	043456	104455			TRAP 55	:	6000
12028	043460	000061			.WORD 61	:	
12029	043462	012706			.WORD ASYNC		
12030	043464	026302			.WORD DUMPER		
12031	043466	012746	012516		MOV #REG.3,-(SP)	:	
12032	043472	012746	010660		MOV #WRD.13,-(SP)	:	6001
12033	043476	012746	010664		MOV #WRD.14,-(SP)		
12034	043502	012746	010730		MOV #WRD.19,-(SP)		
12035	043506	012746	012234		MOV #FNC.7,-(SP)		
12036	043512	012746	012172		MOV #FNC.3,-(SP)		
12037	043516	012746	010322		MOV #SIX.FMT,-(SP)		
12038	043522	012746	000007		MOV #7,-(SP)		
12039	043526	010600			MOV SP,R0	: SP,*	
12040	043530	104414			TRAP 14	:	
12041	043532	062706	000020		ADD #20,SP	:	5999
12042	043536	104467		11\$:	TRAP 67	:	6002
12043	043540	006000			ROR R0	:	
12044	043542	103002			BHIS 12\$		
12045	043544	000167	177164		JMP 1\$		
12046	043550	005726		12\$:	TST (SP)+	:	5934
12047	043552	012601			MOV (SP)+,R1	:	
12048	043554	000207			RTS PC		
12049							
12050							
12051							

: Routine Size: 203 words
: Maximum stack depth per invocation: 17 words

12053
12054
12055
12060
12061
12065
12066
12070
12071
12072
12073
12074
12075
12076
12077
12078
12083
12084
12085 ;

043556
043556 004767 177146
043562 104466
043564 006000
043566 103773
043570 000207

;ML4AD
;
TEST CODE SECTION

T16:: .SBTTL T16 TEST CODE SECTION

1\$: JSR PC,\$T16
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

6006 !<BLF/PAGE>

6004

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 v2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (69)

```

12087 :ML4AD
12088 :
12089 :
12090 : 6007 :
12091 : 6008 : BGNTST;
12092 : 6009 :
12093 : 6010 : !++
12094 : 6011 : TEST NUMBER: TST 17
12095 : 6012 :
12096 : 6013 : TEST NAME: DIAGNOSTIC REGISTER TESTS
12097 : 6014 :
12098 : 6015 : TEST DESCRIPTION:
12099 : 6016 : TEST THE DATA DIAG REGISTERS MLD1, MLD2, MLE2
12100 : 6017 : FOR 1'S/O'S, SHIFTING 1'S/O'S AND INITIALIZATION
12101 : 6018 :
12102 : 6019 : --
12103 : 6020 :
12104 : 6021 : local
12105 : 6022 : CLR_DATA, !CLEAR DATA FOR INIT TEST
12106 : 6023 : SAVE, !TEMPORARY SAVE LOCATION
12107 : 6024 : ERR_FLG, !ERROR FLAG
12108 : 6025 : TST_PAT, !TEST PATTERN
12109 : 6026 : index, !POINTS TO REGISTER PRESENTLY BEING TESTED
12110 : 6027 : DODU_FLG; !DROP UNIT FLAG
12111 : 6028 :
12112 : 6029 : DODU_FLG = ZERO;
12113 : 6030 :
12114 : 6031 :
12115 : 6032 : FIRST TEST THE REGISTERS FOR ONES AND ZEROES
12116 : 6033 :
12117 : 6034 :
12118 : 6035 : TST_PAT = ONES; !LOAD TEST PAT WITH ONES
12119 : 6036 : CLR_THRESHOLD; !CLEAR ERROR PRINT THRESHOLD
12120 : 6037 :
12121 : 6038 : incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
12122 : 6039 : begin
12123 : 6040 :
12124 : 6041 : incr REG_SEL from 11 to 13 do !TEST ELEVEN WRITE/READ REGISTERS
12125 : 6042 : begin
12126 : 6043 : BGNSUB;
12127 : 6044 : CLR_MBUS;
12128 : 6045 : WRT_REG (.TST_PAT, .REG_SEL, index); !WRITE TO THE REGISTER
12129 : 6046 : RD_REG (.TST_PAT, .REG_SEL, ERR_FLG); !READ THE REGISTER
12130 : 6047 :
12131 : 6048 : if .ERR_FLG IS_SET !SEE IF READ FOUND AN ERROR
12132 : 6049 : then
12133 : 6050 : begin
12134 : 6051 : CMP_THRESHOLD; !IF ERROR FLAG IS SET THEN ERROR AND SET DODU_FLG
12135 : 6052 : ERRDF (121, ARR_DAT, DUMPER); !COMPARE ERROR PRINT THRESHOLD
12136 : 6053 : PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56); !ARRAY DATA MODULE FAILURE
12137 : 6054 : PRINTB (FMT_16, .ML_REG [index, REGISTER_ADD], .WT_DATA, .RD_DATA);
12138 : 6055 : DODU_FLG = ONE;
12139 : 6056 : end;
12140 : 6057 :
12141 : 6058 : ENDSUB;

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (69)

```

12143 :ML4AD
12144 :
12145 : TEST CODE SECTION
12146 : 6059 end;
12147 : 6060
12148 : 6061 TST_PAT = not .TST_PAT;
12149 : 6062 end; !REPEAT AGAIN WITH COMPLIMENT DATA
12150 : 6063
12151 : 6064
12152 : 6065
12153 : 6066 TEST THE REGISTERS FOR SHIFTING ONES AND ZEROES
12154 : 6067
12155 : 6068
12156 : 6069 TST_PAT = ONE; !LOAD TST_PAT WITH A 1 IN A FIELD OF 0'S
12157 : 6070
12158 : 6071 incr SHIFT from 0 to 15 do !DO SHIFT 16 TIMES
12159 : 6072 begin
12160 : 6073
12161 : 6074 incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
12162 : 6075 begin
12163 : 6076
12164 : 6077 incr REG_SEL from 11 to 13 do !TEST ELEVEN READ/WRITE REGISTERS
12165 : 6078 begin
12166 : 6079 BGNSUB;
12167 : 6080 CLR_MBUS;
12168 : 6081 WRT_REG (.TST_PAT, .REG_SEL, index); !WRITE TO THE REGISTER
12169 : 6082 RD_REG (.TST_PAT, .REG_SEL, ERR_FLG); !READ THE REGISTER
12170 : 6083
12171 : 6084 if .ERR_FLG IS_SET !SEE IF THE READ FOUND AN ERROR
12172 : 6085 then
12173 : 6086 begin !IF THE ERROR FLAG IS SET THEN ERROR
12174 : 6087 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
12175 : 6088 ERRDF (122, ARR_DAT, DUMPER); !ARRAY DATA MODULE FAILURE
12176 : 6089 PRINTB (SIX_FMT, PHR 4, WRD 12, FNC 5, FNC 6, WRD 52, WRD 56);
12177 : 6090 PRINTB (FMT_16, .ML_REG [.index, REGISTER_ADD], .QT_DATA, .RD_DATA);
12178 : 6091 DODU_FLG = ONE;
12179 : 6092 end;
12180 : 6093
12181 : 6094 ENDSUB;
12182 : 6095 end;
12183 : 6096
12184 : 6097 TST_PAT = not .TST_PAT; !REPEAT WITH A 0 IN A FIELD OF 1'S
12185 : 6098 end;
12186 : 6099
12187 : 6100 TST_PAT = .TST_PAT^ONE; !SHIFT THE 1 IN THE FIELD OF 0'S
12188 : 6101 end;
12189 : 6102
12190 : 6103
12191 : 6104
12192 : 6105 NOW TEST THE REGISTERS FOR INITIALIZATION
12193 : 6106
12194 : 6107
12195 : 6108 REG_INIT_FLG = ONE;
12196 : 6109 TST_PAT = ONES; !BACKGROUND PATTERN
12197 : 6110

```

```

12199 ;ML4AD
12200 :
12201 :
12202 : 6111 incr TWICE from 0 to 1 do
12203 : 6112 begin
12204 : 6113
12205 : 6114 incr REG_SEL from 11 to 13 do
12206 : 6115 begin
12207 : 6116 BGNSUB;
12208 : 6117 CLR_MBUS;
12209 : 6118 WRT_REG (.TST_PAT, .REG_SEL, index);
12210 : 6119 CLR_DATA = (.RI) or (.IGNORE);
12211 : 6120 REG_INIT_FLG = ONE;
12212 : 6121 RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);
12213 : 6122
12214 : 6123 if .ERR_FLG IS_SET
12215 : 6124 then
12216 : 6125 begin
12217 : 6126 CMP THRESHOLD;
12218 : 6127 ERRDF (118, ARR_DAT, DUMPER);
12219 : 6128 PRINTB (FIV_FMT, PHR 4, WRD 12, WRD 52, FNC 23, WRD 56);
12220 : 6129 PRINTB (FMT_16, .ML_REG [index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
12221 : 6130 DODU_FLG = ONE;
12222 : 6131 end;
12223 : 6132
12224 : 6133 ENDSUB;
12225 : 6134 end;
12226 : 6135
12227 : 6136 TST_PAT = not .TST_PAT;
12228 : 6137 end;
12229 : 6138
12230 : 6139 REG_INIT_FLG = ZERO;
12231 : 6140
12232 : 6141 if .DODU_FLG IS_SET
12233 : 6142 then
12234 : 6143 begin
12235 : 6144 DODU (.ML_LUN);
12236 : 6145 DOCLN;
12237 : 6146 end;
12238 : 6147
12239 : 6148 ENDTST;
12240 :
12241 :
12242 :
12243 :
12244 :
12248 043572 004167 141014 $T17: .SBTTL $T17 TEST CODE SECTION
12249 043576 024646 JSR R1,SSAVES
12250 043600 005046 CMP -(SP),-(SP)
12251 043602 012702 CLR -(SP)
12252 043606 005067 MOV #-1,R2
12253 043612 005004 CLR P,CNT
12254 : CLC R4

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (69)

!REPEAT LOOP TWICE
!TEST THIRTEEN REGISTERS
!WRITE REGISTER WITH BACKGROUND
!CALCULATE THE CLEARED DATA PATTERN
!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
!COMPARE ERROR PRINT THRESHOLD
!ARRAY DATA MODULE
!REPEAT WITH COMPLIMENT BACKGROUND PAT
!CLEAR THE FLAG
!DROP THIS UNIT IF DODU_FLG SET
: DODU_FLG
: *.TST.PAT
: TWICE

6005
6029
6035
6038

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Line No.
12311							
12312							
12313							
12314	044074	104467			TRAP	67	
12315	044076	006000			ROR	R0	6056
12316	044100	103647			BLO	2\$	
12317	044102	005201		5\$:	INC	R1	: REG.SEL
12318	044104	020127	000015		CMP	R1,#15	: REG.SEL,*
12319	044110	003643			BLE	2\$	
12320	044112	005102			COM	R2	: TST.PAT
12321	044114	005204			INC	R4	: TWICE
12322	044116	020427	000001		CMP	R4,#1	: TWICE,*
12323	044122	003634			BLE	1\$	
12324	044124	012702	000001		MOV	#1,R2	: *,TST.PAT
12325	044130	005003			CLR	R3	: SHIFT
12326	044132	005004		6\$:	CLR	R4	: TWICE
12327	044134	012701	000013	7\$:	MOV	#13,R1	: *,REG.SEL
12328	044140	104402		8\$:	TRAP	2	
12329	044142	152777	000040	151274	BISB	#40,2ML.REG+40	
12330	044150	016705	151656		MOV	ML,DUT,R5	
12331	044154	042705	177770		BIC	#177770,R5	
12332	044160	142777	000007	151256	BICB	#7,2ML.REG+40	
12333	044166	150577	151252		BISB	R5,2ML.REG+40	
12334	044172	010246			MOV	R2,-(SP)	: TST.PAT,*
12335	044174	010146			MOV	R1,-(SP)	: REG.SEL,*
12336	044176	012746	000012		MOV	#12,-(SP)	
12337	044202	060616			ADD	SP,(SP)	: INDEX,*
12338	044204	004767	161114		JSR	PC,WRT.REG	
12339	044210	010216			MOV	R2,(SP)	: TST.PAT,*
12340	044212	010146			MOV	R1,-(SP)	: REG.SEL,*
12341	044214	012746	000014		MOV	#14,-(SP)	
12342	044220	060616			ADD	SP,(SP)	: ERR.FLG,*
12343	044222	004767	161464		JSR	PC,RD.REG	
12344	044226	026627	000014	000001	CMP	14(SP),#1	: ERR.FLG,*
12345	044234	001065			BNE	10\$	
12346	044236	005267	151134		INC	P.CNT	
12347	044242	026767	151130	151130	CMP	P.CNT,LIMIT	
12348	044250	003403			BLE	9\$	
12349	044252	062706	000012		ADD	#12,SP	
12350	044256	000461			BR	11\$	
12351	044260	104455		9\$:	TRAP	55	
12352	044262	000172			.WORD	172	
12353	044264	013012			.WORD	ARR.DAT	
12354	044266	026302			.WORD	DUMPER	
12355	044270	012746	011344		MOV	#WRD.56,-(SP)	
12356	044274	012746	011310		MOV	#WRD.52,-(SP)	
12357	044300	012746	012226		MOV	#FNC.6,-(SP)	
12358	044304	012746	012216		MOV	#FNC.5,-(SP)	
12359	044310	012746	010650		MOV	#WRD.12,-(SP)	
12360	044314	012746	011676		MOV	#PHR.4,-(SP)	
12361	044320	012746	010322		MOV	#SIX.FMT,-(SP)	
12362	044324	012746	000007		MOV	#7,-(SP)	
12363	044330	010600			MOV	SP,R0	: SP,*
12364	044332	104414			TRAP	14	
12365	044334	016716	151012		MOV	RD.DATA,(SP)	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comment	Page
12423							
12424							
12425							
12426	044604	012746	000014		MOV #14, -(SP)		
12427	044610	060616			ADD SP, (SP)	: ERR.FLG,*	
12428	044612	004767	161074		JSR PC, RD.REG		
12429	044616	026627	000014	000001	CMP 14(SP), #1	: ERR.FLG,*	
12430	044624	001055			BNE 15\$		6123
12431	044626	005267	150544		INC P.CNT	:	
12432	044632	026767	150540	150540	CMP P.CNT, LIMIT		6125
12433	044640	003403			BLE 14\$		
12434	044642	062706	000012		ADD #12, SP		
12435	044646	000451			BR 16\$		
12436	044650	104455			TRAP 55	:	
12437	044652	000166		14\$:	.WORD 166		6127
12438	044654	013012			.WORD ARR.DAT		
12439	044656	026302			.WORD DUMPER		
12440	044660	012746	011344		MOV #WRD.56, -(SP)	:	
12441	044664	012746	012464		MOV #FNC.23, -(SP)		6128
12442	044670	012746	011310		MOV #WRD.52, -(SP)		
12443	044674	012746	010650		MOV #WRD.12, -(SP)		
12444	044700	012746	011676		MOV #PHR.4, -(SP)		
12445	044704	012746	010304		MOV #FIV.FMT, -(SP)		
12446	044710	012746	000006		MOV #6, -(SP)		
12447	044714	010600			MOV SP, R0	: SP,*	
12448	044716	104414			TRAP 14		
12449	044720	016716	150426		MOV RD.DATA, (SP)	:	
12450	044724	010546			MOV R5, -(SP)	: CLR.DATA,*	6129
12451	044726	016146	015404		MOV ML.REG(R1), -(SP)		
12452	044732	012746	007436		MOV #FMT.16, -(SP)		
12453	044736	012746	000004		MOV #4, -(SP)		
12454	044742	010600			MOV SP, R0	: SP,*	
12455	044744	104414			TRAP 14		
12456	044746	012766	000001	000040	MOV #1.40(SP)	: *.DODU.FLG	6130
12457	044754	062706	000026		ADD #26, SP	:	6125
12458	044760	062706	000012		ADD #12, SP	:	6115
12459	044764	104467		15\$:	TRAP 67	:	6131
12460	044766	006000			ROR R0		
12461	044770	103642			BLO 13\$		
12462	044772	005203			INC R3	: REG.SEL	
12463	044774	020327	000015	16\$:	CMP R3, #15	: REG.SEL,*	6114
12464	045000	003636			BLE 13\$		
12465	045002	005102			COM R2	: TST.PAT	6136
12466	045004	005204			INC R4	: TWICE	6111
12467	045006	020427	000001		CMP R4, #1	: TWICE,*	
12468	045012	003627			BLE 12\$		
12469	045014	005067	150340		CLR REG.INIT.FLG	:	6139
12470	045020	021627	000001		CMP (SP), #1	: DODU.FLG,*	6141
12471	045024	001004			BNE 17\$		
12472	045026	016700	150776		MOV ML.LUN, R0	:	6144
12473	045032	104451			TRAP 51		
12474	045034	104444			TRAP 44		
12475	045036	062706	000006	17\$:	ADD #6, SP	:	6005
12476	045042	000207			RTS PC		
12477							

12479
12480
12481
12482
12483
12488
12489
12493
12494
12498 045044
12499 045044 004767 176522
12500 045050 104466
12501 045052 006000
12502 045054 103773
12503 045056 000207
12504
12505
12506
12511
12512
12513 ;

:ML4AD
:
: TEST CODE SECTION
: Routine Size: 341 words
: Maximum stack depth per invocation: 26 words

.SBTTL T17 TEST CODE SECTION
T17::
1\$: JSR PC,\$T17
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

6149 !<BLF/PAGE>

6146

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (70)

```

12515 :ML4AD
12516 :
12517 :
12518 : 6150
12519 : 6151
12520 : 6152
12521 : 6153
12522 : 6154
12523 : 6155
12524 : 6156
12525 : 6157
12526 : 6158
12527 : 6159
12528 : 6160
12529 : 6161
12530 : 6162
12531 : 6163
12532 : 6164
12533 : 6165
12534 : 6166
12535 : 6167
12536 : 6168
12537 : 6169
12538 : 6170
12539 : 6171
12540 : 6172
12541 : 6173
12542 : 6174
12543 : 6175
12544 : 6176
12545 : 6177
12546 : 6178
12547 : 6179
12548 : 6180
12549 : 6181
12550 : 6182
12551 : 6183
12552 : 6184
12553 : 6185
12554 : 6186
12555 : 6187
12556 : 6188
12557 : 6189
12558 : 6190
12559 : 6191
12560 : 6192
12561 : 6193
12562 : 6194
12563 : 6195
12564 : 6196
12565 : 6197
12566 : 6198
12567 : 6199
12568 : 6200
12569 : 6201

TEST CODE SECTION

BGNTST:
!++
TEST NUMBER: TST 18
TEST NAME: COMPOSIT ERROR TEST
TEST DESCRIPTION:
TEST TO SEE IF SETTING EACH
BIT IN THE ERROR REGISTER
CAUSES A COMPOSIT ERROR BY:

WRITING A SHIFTING ONE THROUGH
THE ERROR REGISTER (SKIPPING THE
READ ONLY BITS) AND TESTING THE
COMPOSIT ERROR BIT IN MLDS
FOR BEING SET AFTER EACH
WRITE.

!--
local
  DAT_PAT,
  SKIP_MASK;
!DATA PATTERN
!POINTS TO MLER READ ONLY BITS

CLR MBUS;
SKIP_MASK = %0'163157';
!LOAD SKIP MASK
DAT_PAT = ONE;
!DATA PATTERN SET BIT 0 IN MLER

incr COUNT from 0 to 15 do
begin
!WRITE AND SHIFT DATA PAT TO MLER 16 TIMES

  if (.DAT_PAT and .SKIP_MASK) neq ZERO
  then
!SKIP IF DAT_PAT FALLS ON READ ONLY BIT
    begin
      BGNSUB;
      MLER = .DAT_PAT;
!WRITE DATA_PAT TO MLER

      if .COMP_ERR IS_NOT_SET
      then
!SEE IF DAT_PAT CAUSED A COMP ERROR
        begin
          ERRDF (50, ASYNC, DUMPER);
          PRINTB (FOR_FMT, FNC 8, PHR_1, WRD_12, FNC_8);
!ERROR IF NO COMP ERROR
          PRINTB (FMT_4, .DAT_PAT);
        end;
      end;
    end;
  ENDSUB;
end;

DAT_PAT = .DAT_PAT^ONE;
!SHIFT DAT_PAT TO NEXT BIT AND REPEAT

```


12627
12628
12629
12630
12631
12632
12637
12638
12642
12643
12647 045260
12648 045260 004767 177574
12649 045264 104466
12650 045266 006000
12651 045270 103773
12652 045272 000207
2653
12654
12655
12660
12661
12662 :

:ML4AD
:
TEST CODE SECTION

; Routine Size: 64 words
; Maximum stack depth per invocation: 12 words

.SBTTL T18 TEST CODE SECTION
T18::
1\$: JSR PC,ST18
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

6205 !<BLF/PAGE>

6202

29-Mar-1982 16:23:04
29-Mar-1982 16:27:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (71)

12664 :ML4AD
12665 :
12666 :
12667 :
12668 :
12669 :
12670 :
12671 :
12672 :
12673 :
12674 :
12675 :
12676 :
12677 :
12678 :
12679 :
12680 :
12681 :
12682 :
12683 :
12684 :
12685 :
12686 :
12687 :
12688 :
12689 :
12690 :
12691 :
12692 :
12693 :
12694 :
12695 :
12696 :
12697 :
12698 :
12699 :
12700 :
12701 :
12702 :
12703 :
12704 :
12705 :
12706 :
12707 :
12708 :
12709 :
12710 :
12711 :
12712 :
12713 :
12714 :
12715 :
12716 :
12717 :
12718 :

6206
6207
6208
6209
6210
6211
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
6224
6225
6226
6227
6228
6229
6230
6231
6232
6233
6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257

```
TEST CODE SECTION
BGNTST:
!++
TEST NUMBER: TST 19
TEST NAME: ATA BIT TEST
TEST DESCRIPTION:
TEST THE ATA BIT FOR SETTING
BY:
1. SETTING A BIT IN THE ERROR
REGISTER.
TEST THE ATA BIT FOR CLEARING
AFTER BEING SET BY:
1. WRITING A FUNCTION TO MLCS1.
2. WRITING A ONE INTO THIS
UNITS ATA BIT
TEST THE ATA BIT FOR NOT CLEARING
AFTER BEING SET BY:
1. WRITING A ONE INTO THE
ATA BIT OF THE OTHER
UNITS.
--
local
ATA_SAVE : bitvector [8],
DAT_PAT;
CLR MBUS;
MLER = ONE;
MLER = ZERO;
ATA_SAVE = .MLAS;
if .ATA_SAVE [.ML_DUT] IS_NOT_SET
then
begin
ERRDF (51, ASYNC, DUMPER);
PRINTB (FOR_FMT, WRD_15, PHR_1, WRD_11, FNC_8);
EXIT_TST;
end;
if .ATTN IS_NOT_SET
then
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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (71)

```

12720 :ML4AD
12721 :
12722 : TEST CODE SECTION
12723 : 6258 ERRDF (52, ASYNC, DUMPER); !ERROR AND EXIT_TST IF NOT SET
12724 : 6259 PRINTB (FIV_FMT, REG_2, WRD_16, PHR_1, WRD_11, FNC_8);
12725 : 6260 EXIT_TST;
12726 : 6261 end;
12727 : 6262
12728 : 6263 MLCS1 = NOOP; !TRY TO CLEAR THE ATA BIT WITH NOOP FUNC
12729 : 6264
12730 : 6265 if .ATTN IS_SET !SEE IF ATA GOT CLEARED
12731 : 6266 then
12732 : 6267 begin
12733 : 6268 ERRDF (53, ASYNC, DUMPER); !ERROR AND EXIT_TST IF SET
12734 : 6269 PRINTB (FOR_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
12735 : 6270 EXIT_TST;
12736 : 6271 end;
12737 : 6272
12738 : 6273 ATA_SAVE = .MLAS; !READ THE ATTENTION REGISTER
12739 : 6274
12740 : 6275 if .ATA_SAVE [.ML_DUT] IS_SET !SEE IF THE ATA REG GOT CLEARED BY NO-OP
12741 : 6276 then
12742 : 6277 begin
12743 : 6278 ERRDF (58, ASYNC, DUMPER);
12744 : 6279 PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, FNC_2, WRD_19);
12745 : 6280 end;
12746 : 6281
12747 : 6282 BGNSUB;
12748 : 6283 MLER = ONE; !SET THE ATA BIT
12749 : 6284 MLER = ZERO;
12750 : 6285 ATA_SAVE = ZEROES; !CLEAR ATA_SAVE
12751 : 6286 ATA_SAVE [.ML_DUT] = ONE; !SET ATA_SAVE FOR THIS DRIVE
12752 : 6287 MLAS = .ATA_SAVE; !TRY TO CLEAR THE ATA BY WRITING TO IT.
12753 : 6288
12754 : 6289 if .ATTN IS_SET !SEE IF THE ATA GOT CLEARED
12755 : 6290 then
12756 : 6291 begin
12757 : 6292 ERRDF (54, ASYNC, DUMPER); !ERROR IF NOT CLEARED
12758 : 6293 PRINTB (FIV_FMT, WRD_15, PHR_2, WRD_11, WRD_17, REG_5);
12759 : 6294 end;
12760 : 6295
12761 : 6296 ENDSUB;
12762 : 6297 BGNSUB;
12763 : 6298 MLER = ONE; !SET THE ATA BIT
12764 : 6299 MLER = ZERO;
12765 : 6300 DAT_PAT = ONE; !DATA PATTERN OF ONE IN FIELD OF ZEROES
12766 : 6301
12767 : 6302 incr ATA_SEL from 0 to 7 do !REPEAT LOOP 8 TIMES
12768 : 6303 begin
12769 : 6304
12770 : 6305 if .ATA_SEL neq .ML_DUT !SKIP IF ATA_SEL EQLS THIS DRIVE NO.
12771 : 6306 then
12772 : 6307 begin
12773 : 6308 MLAS = .DAT_PAT; !WRITE DAT PAT TO ATA REGISTER
12774 : 6309 ATA_SAVE = .MLAS; !READ ATA REG BACK

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (71)

12776 :ML4AD
12777 :
12778 :
12779 : 6310
12780 : 6311
12781 : 6312
12782 : 6313
12783 : 6314
12784 : 6315
12785 : 6316
12786 : 6317
12787 : 6318
12788 : 6319
12789 : 6320
12790 : 6321
12791 : 6322
12792 : 6323
12793 : 6324
12794 : 6325
12795 : 6326
12799 :

TEST CODE SECTION

```
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;
```

12800
12804 045274 004167 137254
12805 045300 152777 000040 150136
12806 045306 016703 150520
12807 045312 042703 177770
12808 045316 142777 000007 150120
12809 045324 150377 150114
12810 045330 012777 000001 150126
12811 045336 005077 150122
12812 045342 017746 150126
12813 045346 016701 150460
12814 045352 006201
12815 045354 006201
12816 045356 006201
12817 045360 010600
12818 045362 060001
12819 045364 010146
12820 045366 016746 150440
12821 045372 042716 177770
12822 045376 012746 000001
12823 045402 005046
12824 045404 004767 136224
12825 045410 062706 000010
12826 045414 005700
12827 045416 001026
12828 045420 104455
12829 045422 000063
12830 045424 012706

```
.SBTTL $T19 TEST CODE SECTION
$T19: JSR R1,$SAVE3
      BISB #40,@ML.REG+40
      MOV ML_DUT,R3
      BIC #177770,R3
      BICB #7,@ML.REG+40
      BISB R3,@ML.REG+40
      MOV #1,@ML.REG+60
      CLR @ML.REG+60
      MOV @ML.REG+70,-(SP)
      MOV ML_DUT,R1
      ASR R1
      ASR R1
      ASR R1
      MOV SP,R0
      ADD R0,R1
      MOV R1,-(SP)
      MOV ML_DUT,-(SP)
      BIC #177770,(SP)
      MOV #1,-(SP)
      CLR -(SP)
      JSR PC,BLSGT2
      ADD #10,SP
      TST R0
      BNE 1$
      TRAP 55
      .WORD 63
      .WORD ASYNC
```

6204
6240
6243
6244
6245
6247
6250

Address	Hex	Hex	Hex	Label	Code	Comment	Address	
12832								
12833								
12834								
12835	045426	026302			.WORD	DUMPER		
12836	045430	012746	012244		MOV	#FNC.8,-(SP)		
12837	045434	012746	010640		MOV	#WRD.11,-(SP)	6251	
12838	045440	012746	011610		MOV	#PHR.1,-(SP)		
12839	045444	012746	010676		MOV	#WRD.15,-(SP)		
12840	045450	012746	010270		MOV	#FOR.FMT,-(SP)		
12841	045454	012746	000005		MOV	#5,-(SP)		
12842	045460	010600			MOV	SP,R0	: SP,*	
12843	045462	104414			TRAP	14		
12844	045464	104463			TRAP	63		
12845	045466	062706	000014		ADD	#14,SP		
12846	045472	000467			BR	4\$	6247	
12847	045474	032777	100000	147752	1\$:	BIT	#100000,@ML.REG+50	6249
12848	045502	001026			BNE	2\$	6255	
12849	045504	104455			TRAP	55		
12850	045506	000064			.WORD	64	6258	
12851	045510	012706			.WORD	ASYN		
12852	045512	026302			.WORD	DUMPER		
12853	045514	012746	012244		MOV	#FNC.8,-(SP)		
12854	045520	012746	010640		MOV	#WRD.11,-(SP)	6259	
12855	045524	012746	011610		MOV	#PHR.1,-(SP)		
12856	045530	012746	010704		MOV	#WRD.16,-(SP)		
12857	045534	012746	012510		MOV	#REG.2,-(SP)		
12858	045540	012746	010304		MOV	#FIV.FMT,-(SP)		
12859	045544	012746	000006		MOV	#6,-(SP)		
12860	045550	010600			MOV	SP,R0	: SP,*	
12861	045552	104414			TRAP	14		
12862	045554	104463			TRAP	63		
12863	045556	000433			BR	3\$		
12864	045560	012777	000001	147616	2\$:	MOV	#1,@ML.REG	6255
12865	045566	005777	147662		TST	@ML.REG+50	6263	
12866	045572	100031			BPL	5\$	6265	
12867	045574	104455			TRAP	55		
12868	045576	000065			.WORD	65	6268	
12869	045600	012706			.WORD	ASYN		
12870	045602	026302			.WORD	DUMPER		
12871	045604	012746	010730		MOV	#WRD.19,-(SP)		
12872	045610	012746	012164		MOV	#FNC.2,-(SP)	6269	
12873	045614	012746	010640		MOV	#WRD.11,-(SP)		
12874	045620	012746	011626		MOV	#PHR.2,-(SP)		
12875	045624	012746	010676		MOV	#WRD.15,-(SP)		
12876	045630	012746	010270		MOV	#FOR.FMT,-(SP)		
12877	045634	012746	000006		MOV	#6,-(SP)		
12878	045640	010600			MOV	SP,R0	: SP,*	
12879	045642	104414			TRAP	14		
12880	045644	104463			TRAP	63		
12881	045646	062706	000016		ADD	#16,SP		
12882	045652	000167	000546		JMP	12\$	6265	
12883	045656	017716	147612		MOV	@ML.REG+70,(SP)	6267	
12884	045662	016701	150144		MOV	ML.DUT,R1	6273	
12885	045666	006201			ASR	R1	6275	
12886	045670	006201			ASR	R1		

Address	Hex	Hex	Hex	Label	Instruction	Comments	Address
12888							
12889							
12890							
12891	045672	006201			ASR R1		
12892	045674	010600			MOV SP,R0		
12893	045676	060001			ADD R0,R1	: ATA.SAVE,*	
12894	045700	010146			MOV R1,-(SP)		
12895	045702	016746	150124		MOV ML.DUT,-(SP)		
12896	045706	042716	177770		BIC #177770,(SP)		
12897	045712	012746	000001		MOV #1,-(SP)		
12898	045716	005046			CLR -(SP)		
12899	045720	004767	135710		JSR PC,BLSGT2		
12900	045724	062706	000010		ADD #10,SP		
12901	045730	005300			DEC R0		
12902	045732	001026			BNE 6\$		
12903	045734	104455			TRAP 55		
12904	045736	000072			.WORD 72	:	6278
12905	045740	012706			.WORD ASYNC		
12906	045742	026302			.WORD DUMPER		
12907	045744	012746	010730		MOV #WRD.19,-(SP)	:	
12908	045750	012746	012164		MOV #FNC.2,-(SP)	:	6279
12909	045754	012746	010640		MOV #WRD.11,-(SP)		
12910	045760	012746	011626		MOV #PHR.2,-(SP)		
12911	045764	012746	010676		MOV #WRD.15,-(SP)		
12912	045770	012746	010304		MOV #FIV.FMT,-(SP)		
12913	045774	012746	000006		MOV #6,-(SP)		
12914	046000	010600			MOV SP,R0	: SP,*	
12915	046002	104414			TRAP 14	:	
12916	046004	062706	000016		ADD #16,SP	:	
12917	046010	104402			TRAP 2	:	6277
12918	046012	012777	000001	147444	MOV #1,@ML.REG+60	:	6280
12919	046020	005077	147440		CLR @ML.REG+60	:	6283
12920	046024	005016			CLR (SP)	:	6284
12921	046026	016701	150000		MOV ML.DUT,R1	: ATA.SAVE	6285
12922	046032	006201			ASR R1	:	6286
12923	046034	006201			ASR R1		
12924	046036	006201			ASR R1		
12925	046040	010600			MOV SP,R0	: ATA.SAVE,*	
12926	046042	060001			ADD R0,R1		
12927	046044	010146			MOV R1,-(SP)		
12928	046046	016746	147760		MOV ML.DUT,-(SP)		
12929	046052	042716	177770		BIC #177770,(SP)		
12930	046056	012746	000001		MOV #1,-(SP)		
12931	046062	011646			MOV (SP),-(SP)		
12932	046064	004767	136002		JSR PC,BLSPU2		
12933	046070	016677	000010	147376	MOV 10(SP),@ML.REG+70	: ATA.SAVE,*	
12934	046076	005777	147352		TST @ML.REG+50	:	6287
12935	046102	100026			BPL 7\$:	6289
12936	046104	104455			TRAP 55	:	
12937	046106	000066			.WORD 66	:	6292
12938	046110	012706			.WORD ASYNC		
12939	046112	026302			.WORD DUMPER		
12940	046114	012746	012532		MOV #REG.5,-(SP)	:	
12941	046120	012746	010712		MOV #WRD.17,-(SP)		6293
12942	046124	012746	010640		MOV #WRD.11,-(SP)		

Address	Hex	Hex	Hex	Label	Code	Comments	Line No.
12944				:ML4AD			
12945				:			
12946					TEST CODE SECTION		
12947	046130	012746	011626		MOV #PHR.2, -(SP)		
12948	046134	012746	010676		MOV #WRD.15, -(SP)		
12949	046140	012746	010304		MOV #FIV.FMT, -(SP)		
12950	046144	012746	000006		MOV #6, -(SP)		
12951	046150	010600			MOV SP, R0		
12952	046152	104414			TRAP 14	: SP, *	
12953	046154	062706	000016		ADD #16, SP	:	6291
12954	046160	062706	000010	7\$:	ADD #10, SP	:	6280
12955	046164	104467			TRAP 67	:	6294
12956	046166	006000			ROR R0	:	
12957	046170	103707			BLO 6\$:	
12958	046172	104402		8\$:	TRAP 2	:	
12959	046174	012777	000001	147262	MOV #1, @ML.REG+60	:	6296
12960	046202	005077	147256		CLR @ML.REG+60	:	6299
12961	046206	012703	000001		MOV #1, R3	:	6299
12962	046212	005002			CLR R2	: *, DAT.PAT	6300
12963	046214	020267	147612	9\$:	CMP R2, ML.DUT	: ATA.SEL	6302
12964	046220	001471			BEQ 10\$: ATA.SEL, *	6305
12965	046222	010377	147246		MOV R3, @ML.REG+70	: DAT.PAT, *	6308
12966	046226	017716	147242		MOV @ML.REG+70, (SP)	: *, ATA.SAVE	6309
12967	046232	016701	147574		MOV ML.DUT, R1	:	6311
12968	046236	006201			ASR R1	:	
12969	046240	006201			ASR R1	:	
12970	046242	006201			ASR R1	:	
12971	046244	010600			MOV SP, R0	: ATA.SAVE, *	
12972	046246	060001			ADD R0, R1	:	
12973	046250	010146			MOV R1, -(SP)	:	
12974	046252	016746	147554		MOV ML.DUT, -(SP)	:	
12975	046256	042716	177770		BIC #177770, (SP)	:	
12976	046262	012746	000001		MOV #1, -(SP)	:	
12977	046266	005046			CLR -(SP)	:	
12978	046270	004767	135340		JSR PC, BL\$GT2	:	
12979	046274	062706	000010		ADD #10, SP	:	
12980	046300	005700			TST R0	:	
12981	046302	001040			BNE 10\$:	
12982	046304	104455			TRAP 55	:	6314
12983	046306	000067			.WORD 67	:	
12984	046310	012706			.WORD ASYNC	:	
12985	046312	026302			.WORD DUMPER	:	
12986	046314	012746	011740		MOV #PHR.7, -(SP)	:	6315
12987	046320	012746	012532		MOV #REG.5, -(SP)	:	
12988	046324	012746	010712		MOV #WRD.17, -(SP)	:	
12989	046330	012746	010640		MOV #WRD.11, -(SP)	:	
12990	046334	012746	011726		MOV #PHR.6, -(SP)	:	
12991	046340	012746	010676		MOV #WRD.15, -(SP)	:	
12992	046344	012746	010322		MOV #SIX.FMT, -(SP)	:	
12993	046350	012746	000007		MOV #7, -(SP)	:	
12994	046354	010600			MOV SP, R0	: SP, *	
12995	046356	104414			TRAP 14	:	
12996	046360	010316			MOV R3, (SP)	: DAT.PAT, *	6316
12997	046362	012746	007002		MOV #FMT.7, -(SP)	:	
12998	046366	012746	000002		MOV #2, -(SP)	:	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

13000
13001
13002
13003 046372 010600
13004 046374 104414
13005 046376 062706 000024
13006 046402 000405
13007 046404 006303
13008 046406 005202
13009 046410 020227 000007
13010 046414 003677
13011 046416 104467
13012 046420 006000
13013 046422 103663
13014 046424 005726
13015 046426 000207

:ML4AD
:
TEST CODE SECTION

MOV SP,R0 : SP,*
TRAP 14 :
ADD #24,SP :
BR 11\$:
10\$: ASL R3 : DAT.PAT
INC R2 : ATA.SEL
CMP R2,#7 : ATA.SEL,*
BLE 9\$:
11\$: TRAP 67 :
ROR R0 :
BLO 8\$:
12\$: TST (SP)+ :
RTS PC :

6317

6322

6302

6323

6204

: Routine Size: 302 words
: Maximum stack depth per invocation: 16 words

13016
13017
13018
13023
13024
13028
13029
13033 046430
13034 046430 004767 176640
13035 046434 104466
13036 046436 006000
13037 046440 103773
13038 046442 000207

.SBTTL T19 TEST CODE SECTION

T19::
1\$: JSR PC,\$T19 :
TRAP 66 :
ROR R0 :
BLO 1\$:
RTS PC :

6325

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

13039
13040
13041
13046
13047
13048 : 6327 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLi.4 (72)

13050 :ML4AD
13051 :
13052 :
13053 :
13054 :
13055 :
13056 :
13057 :
13058 :
13059 :
13060 :
13061 :
13062 :
13063 :
13064 :
13065 :
13066 :
13067 :
13068 :
13069 :
13070 :
13071 :
13072 :
13073 :
13074 :
13075 :
13076 :
13077 :
13078 :
13079 :
13080 :
13081 :
13082 :
13083 :
13084 :
13085 :
13086 :
13087 :
13088 :
13089 :
13090 :
13091 :
13092 :
13093 :
13094 :
13095 :
13096 :
13097 :
13098 :
13099 :
13100 :
13101 :
13102 :
13103 :
13104 :

6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376
6377
6378
6379

TEST CODE SECTION

EGNTST;

!++

TEST NUMBER: TST 20
TEST NAME: SEARCH FUNCTION TEST
TEST DESCRIPTION:

TEST THE SEARCH FUNCTION BY:

1. DOING A SEARCH FUNCTION AT ARRAY ZERO AND TEST GO, ERROR BITS AND ATTN FOR SETTING/NOT SETTING.
2. DOING SEARCH FUNCTIONS AT ALL PRESENT ARRAYS' AND TEST ATTN SET
3. DOING SEARCH FUNCTIONS AT ALL NOT PRESENT ARRAYS' AND TEST ATTN CLEARED.

--

CLR MBUS;

MLDA = ZEROES;

MLCS1 = SEARCH;

!DO A SEARCH FUNCTION

if .GO IS_SET

!SEE IF GO IS SET

then

begin

ERRDF (56, ASYNC, DUMPER);

!ERROR IF NOT SET

PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_10, WRD_19);

end;

if .ILF IS_SET

!SEE IF ILF IS SET

then

begin

ERRDF (57, ASYNC, DUMPER);

!ERROR IF SET

PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, FNC_10, WRD_19);

end;

if .ATTN IS_NOT_SET

!SEE IF ATTN IS SET

then

begin

ERRDF (59, ASYNC, DUMPER);

!ERROR IF NOT SET

PRINTB (FIV_FMT, WRD_16, PHR_1, WRD_12, FNC_10, WRD_19);

end

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (72)

```

13106 :ML4AD
13107 :
13108 :
13109 : 6380 else
13110 : 6381   begin
13111 : 6382
13112 : 6383   CLR_THRESHOLD;
13113 : 6384
13114 : 6385
13115 : 6386   !! VERSION CZMLAD CHANGED INCR TO INCRU
13116 : 6387   !!
13117 : 6388   incru ARR_SEL from 0 to .LST_ARR by .ARR_INC do
13118 : 6389     begin
13119 : 6390     BGNSUB;
13120 : 6391     CLR MBUS;
13121 : 6392     MLDA = .ARR_SEL;
13122 : 6393     MLCS1 = SEARCH;
13123 : 6394
13124 : 6395     if .OPI IS_SET
13125 : 6396     then
13126 : 6397       begin
13127 : 6398       CMP THRESHOLD;
13128 : 6399       ERRDF (60, ASYNC, DUMPER);
13129 : 6400       PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_12, FNC_10, WRD_19);
13130 : 6401       PRINTB (FMT_9, .ARR_SEL);
13131 : 6402     end;
13132 : 6403
13133 : 6404     ENDSUB;
13134 : 6405     end;
13135 : 6406
13136 : 6407   CLR_THRESHOLD;
13137 : 6408
13138 : 6409   if .OP_NUM_ARR lss %o'000017'
13139 : 6410   then
13140 : 6411
13141 : 6412   !!
13142 : 6413   !! VERSION CZMLAD CHANGED INCR TO INCRU
13143 : 6414   !!
13144 : 6415   incru ARR_SEL from (.LST_ARR + .ARR_INC) to .ARR_16 by .ARR_INC do
13145 : 6416     begin
13146 : 6417     BGNSUB;
13147 : 6418     CLR MBUS;
13148 : 6419     MLDA = .ARR_SEL;
13149 : 6420     MLCS1 = SEARCH;
13150 : 6421
13151 : 6422     if .OPI IS_NOT_SET
13152 : 6423     then
13153 : 6424       begin
13154 : 6425       CMP THRESHOLD;
13155 : 6426       ERRDF (61, ASYNC, DUMPER);
13156 : 6427       PRINTB (FIV_FMT, WRD_4, PHR_1, WRD_12, FNC_10, WRD_19);
13157 : 6428       PRINTB (FMT_9, .ARR_SEL);
13158 : 6429     end;
13159 : 6430
13160 : 6431
  
```


29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (72)

13162 :ML4AD
13163 :
13164 :
13165 :
13166 :
13167 :
13168 :
13169 :
13170 :
13174 :
13175 :

TEST CODE SECTION

6432 ENDSUB:
6433 end:
6434
6435 end:
6436
6437 ENDTST:

13179 046444 004167 136104
13180 046450 152777 000040 146766
13181 046456 016703 147350
13182 046462 042703 177770
13183 046466 142777 000007 146750
13184 046474 150377 146744
13185 046500 005077 146730
13186 046504 012777 000031 146672
13187 046512 132777 000001 146664
13188 046520 001426
13189 046522 104455
13190 046524 000070
13191 046526 012706
13192 046530 026302
13193 046532 012746 010730
13194 046536 012746 012272
13195 046542 012746 010640
13196 046546 012746 011626
13197 046552 012746 010520
13198 046556 012746 010304
13199 046562 012746 000006
13200 046566 010600
13201 046570 104414
13202 046572 062706 000016
13203 046576 132777 000001 146660 1S:
13204 046604 001430
13205 046606 104455
13206 046610 000071
13207 046612 012706
13208 046614 026302
13209 046616 012746 010730
13210 046622 012746 012272
13211 046626 012746 012302
13212 046632 012746 010640
13213 046636 012746 011714
13214 046642 012746 010536
13215 046646 012746 010304
13215 046652 012746 000007

\$T20:

.SBTTL \$T20 TEST CODE SECTION

JSR R1,\$SAVE3
BISB #40,@ML.REG+40
MOV ML.DUT,R3
BIC #177770,R3
BICB #7,@ML.REG+40
BISB R3,@ML.REG+40
CLR @ML.REG+30
MOV #31,@ML.REG
BITB #1,@ML.REG
BEQ 1S
TRAP 55
.WORD 70
.WORD ASYNC
.WORD DUMPER
MOV #WRD.19,-(SP)
MOV #FNC.10,-(SP)
MOV #WRD.11,-(SP)
MOV #PHR.2,-(SP)
MOV #WRD.1,-(SP)
MOV #FIV.FMT,-(SP)
MOV #6,-(SP)
MOV SP,R0
TRAP 14 ; SP,*
ADD #16,SP
BITB #1,@ML.REG+60
BEQ 2S
TRAP 55
.WORD 71
.WORD ASYNC
.WORD DUMPER
MOV #WRD.19,-(SP)
MOV #FNC.10,-(SP)
MOV #FNC.11,-(SP)
MOV #WRD.11,-(SP)
MOV #PHR.5,-(SP)
MOV #WRD.3,-(SP)
MOV #FIV.FMT,-(SP)
MOV #7,-(SP)

6326
6330
6357
6358
6360
6363
6364
6362
6367
6370
6371

Address	Hex	Hex	Hex	Label	Instruction	Comments	Line No.
13218							
13219				:ML4AD			
13220				:	TEST CODE SECTION		
13221	046656	010600			MOV SP,R0	: SP,*	
13222	046660	104414			TRAP 14	:	
13223	046662	062706	000020		ADD #20,SP	:	
13224	046666	032777	100000	146560 2\$:	BIT #100000,@ML.REG+50	:	6369
13225	046674	001027			BNE 3\$:	6374
13226	046676	104455			TRAP 55	:	
13227	046700	000073			.WORD 73	:	6377
13228	046702	012706			.WORD ASYNC	:	
13229	046704	026302			.WORD DUMPER	:	
13230	046706	012746	010730		MOV #WRD.19,-(SP)	:	
13231	046712	012746	012272		MOV #FNC.10,-(SP)	:	6378
13232	046716	012746	010650		MOV #WRD.12,-(SP)	:	
13233	046722	012746	011610		MOV #PHR.1,-(SP)	:	
13234	046726	012746	010704		MOV #WRD.16,-(SP)	:	
13235	046732	012746	010304		MOV #FIV.FMT,-(SP)	:	
13236	046736	012746	000006		MOV #6,-(SP)	:	
13237	046742	010600			MOV SP,R0	: SP,*	
13238	046744	104414			TRAP 14	:	
13239	046746	062706	000016		ADD #16,SP	:	
13240	046752	000207			RTS PC	:	6376
13241	046754	005067	146416	3\$:	CLR P.CNT	:	6374
13242	046760	016702	145034		MOV LST.ARR,R2	:	6381
13243	046764	016703	145014		MOV ARR.INC,R3	:	6388
13244	046770	005001			CLR R1	: ARR.SEL	
13245	046772	000475			BR 7\$:	
13246	046774	104402		4\$:	TRAP 2	:	
13247	046776	152777	000040	146440	BISB #40,@ML.REG+40	:	6389
13248	047004	016700	147022		MOV ML.DUT,R0	:	6390
13249	047010	042700	177770		BIC #177770,R0	:	
13250	047014	142777	000007	146422	BICB #7,@ML.REG+40	:	
13251	047022	150077	146416		BISB R0,@ML.REG+40	:	
13252	047026	010177	146402		MOV R1,@ML.REG+30	: ARR.SEL,*	6392
13253	047032	012777	000031	146344	MOV #31,@ML.REG	:	6393
13254	047040	032777	020000	146416	BIT #20000,@ML.REG+60	:	6395
13255	047046	001443			BEQ 5\$:	
13256	047050	005267	146322		INC P.CNT	:	6397
13257	047054	026767	146316	146316	CMP P.CNT,LIMIT	:	
13258	047062	003040			BGT 6\$:	
13259	047064	104455			TRAP 55	:	6399
13260	047066	000074			.WORD 74	:	
13261	047070	012706			.WORD ASYNC	:	
13262	047072	026302			.WORD DUMPER	:	
13263	047074	012746	010730		MOV #WRD.19,-(SP)	:	
13264	047100	012746	012272		MOV #FNC.10,-(SP)	:	6400
13265	047104	012746	010650		MOV #WRD.12,-(SP)	:	
13266	047110	012746	011714		MOV #PHR.5,-(SP)	:	
13267	047114	012746	010544		MOV #WRD.4,-(SP)	:	
13268	047120	012746	010304		MOV #FIV.FMT,-(SP)	:	
13269	047124	012746	000006		MOV #6,-(SP)	:	
13270	047130	010600			MOV SP,R0	: SP,*	
13271	047132	104414			TRAP 14	:	
13272	047134	010116			MOV R1,(SP)	: ARR.SEL,*	6401

13330
 13331
 13332
 13333 047414 006000
 13334 047416 103704
 13335 047420 060201
 13336 047422 020103
 13337 047424 101701
 13338 047426 000207
 13339
 13340
 13341
 13346
 13347
 13351
 13352
 13356 047430
 13357 047430 004767 177010
 13358 047434 104466
 13359 047436 006000
 13360 047440 103773
 13361 047442 000207
 13362
 13363
 13364
 13369
 13370
 13371 ;

:ML4AD
 :
 : TEST CODE SECTION

ROR R0
 BLO 8\$
 10\$: ADD R2,R1 : *,ARR.SEL
 11\$: CMP R1,R3 : ARR.SEL,*
 BLOS 8\$
 12\$: RTS PC :
 :
 : Routine Size: 250 words
 : Maximum stack depth per invocation: 13 words

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

6415
 6326

.SBTTL T20 TEST CODE SECTION
 T20::
 1\$: JSR PC,ST20 :
 TRAP 66 :
 ROR R0
 BLO 1\$
 RTS PC

6435

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

6438 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<F> <LE>ML4AD.BLI.4 (73)

13373 :ML4AD
13374 :
13375 :
13376 :
13377 :
13378 :
13379 :
13380 :
13381 :
13382 :
13383 :
13384 :
13385 :
13386 :
13387 :
13388 :
13389 :
13390 :
13391 :
13392 :
13393 :
13394 :
13395 :
13396 :
13397 :
13398 :
13399 :
13400 :
13401 :
13402 :
13403 :
13404 :
13405 :
13406 :
13407 :
13408 :
13409 :
13410 :
13411 :
13412 :
13413 :
13414 :
13415 :
13416 :
13417 :
13418 :
13419 :
13420 :
13421 :
13422 :
13423 :
13424 :
13425 :
13426 :
13427 :

TEST CODE SECTION

6439 :
6440 :BGNTST;
6441 :
6442 :++

6443 :TEST NUMBER: TST 21

6444 :
6445 :TEST NAME: POWER FAIL TEST

6446 :
6447 :TEST DESCRIPTION:

6448 :THIS TEST IS PART OF THE MANUAL INTERVENTION
6449 :TEST SECTION. DRIVE CONDITIONS REVOLVING
6450 :AROUND POWER FAIL SITUATION ARE TEST FOR.
6451 :FUNCTION TESTED FOR ARE AS FOLLOWS:

- 6452 : 1. VV BIT SETTING AND CLEARING
- 6453 : 2. READ-IN-PRESET FUNCTION SETTING
- 6454 : THE VV BIT.
- 6455 : 3. ECC INITIALIZE CLEARING OUT MEMORY
- 6456 : WITH BAD BATTERY BACK-UP.
- 6457 : 4. DRIVE REGISTERS BEING CLEARED BY
- 6458 : DC LO via 'MB_DIS LO'.
- 6459 : 5. UNSAFE BIT 'UNS' SETTING DURING AC
- 6460 : POWER LOSS.

6461 :
6462 :--
6463 :
6464 :Local

6465 :SIZE,
6466 :DST,
6467 :SRC,
6468 :DSA_CNT,
6469 :ERR_FLG,
6470 :CLR_DATA,
6471 :index,
6472 :WCE_CNT;

!STORES THE SIZE OF MASS BUS TRANSFERS
!STORES THE DESTINATION ADRS OF MASS BUS TRANSFERS
!STORES THE SOURCE ADRS OF MASS BUS TRANSFERS
!COUNTS THE NUMBER OF BLOCK TRANSFERS DONE
!ERROR FLAG
!STORES THE EXPECTED CLEARED DATA FROM THE REGISTERS
!INDEX VALUE INTO ML REG STRUCTURE TO ACCESS THE REGISTERS ADRS
!COUNT THE NUMBER OF WRITE CHECK ERROR DETECTED

6473 :+
6474 :PERFORM THIS TEST ONLY IF THE OPERATOR HAS
6475 :ENABLED MANUAL INTERVENTION TEST VIA THE S/W
6476 :QUESTIONS.
6477 :--

6478 :if .INTERVEN
6479 :then
6480 :begin

!DO THIS TEST IF INTERVEN IS SET

6481 :
6482 :PRINT WHICH TEST THIS IS
6483 :
6484 :
6485 :
6486 :
6487 :
6488 :
6489 :
6490 :

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (73)

13429 :ML4AD
13430 :
13431 :
13432 :
13433 :
13434 :
13435 :
13436 :
13437 :
13438 :
13439 :
13440 :
13441 :
13442 :
13443 :
13444 :
13445 :
13446 :
13447 :
13448 :
13449 :
13450 :
13451 :
13452 :
13453 :
13454 :
13455 :
13456 :
13457 :
13458 :
13459 :
13460 :
13461 :
13462 :
13463 :
13464 :
13465 :
13466 :
13467 :
13468 :
13469 :
13470 :
13471 :
13472 :
13473 :
13474 :
13475 :
13476 :
13477 :
13478 :
13479 :
13480 :
13481 :
13482 :
13483 :

TEST CODE SECTION

```
PRINTB (ONE_FMT, T_21);
BGNSUB;
```

```
!PRINT T 21 MESSAGE
!START SCOOP LOOP
```

```
FIRST TEST THE VV BIT FOR SETTING.
TO DO THIS FIRST SEE IF THE BIT IS
ALREADY SET. IF ALREADY SET THEN
SEE IF THE READ-IN-PRESET FUNCTION
CLEARS THE BIT. IF NOT ALREADY SET
THEN SEE IF THE READ-IN-PRESET FUNC
SETS THE BIT.
```

```
CLR_MBUS;
```

```
!CLEAR THE MASS BUS
```

```
if .VV
then
begin
MLCS1 = RD_IN_PRE;
```

```
!IS THE VV BIT INITIALLY SET
!THE VV BIT IS SET
!SEE IF THE READ-IN-PRESET FUNC CLEARS IT
!DO A READ IN PRESET FUNCTION
```

```
if not (.VV)
then
begin
ERRDF (149, ASYNC, DUMPER);
PRINTB (ONE_FMT, VV_CLEAR);
end
```

```
!DID THE VV GET CLEARED
!THE VV GOT CLEARED
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS
```

```
end
else
begin
MLCS1 = RD_IN_PRE;
```

```
!THE VV BIT IS INITIALLY CLEARED
!SEE IF A READ-IN-PRESET FUNC SETS THE VV BIT
!DO A READ-IN-PRESET FUNCTION
```

```
if not (.VV)
then
begin
ERRDF (150, ASYNC, DUMPER);
PRINTB (TWO_FMT, VV_NOT_SET, FNC_11);
end;
```

```
!DID THE VV BIT GET SET
!THE VV BIT DID NOT SET
!REPORT THE ERROR TO OPERATOR
!TELL WHAT THE ERROR IS
```

```
end;
```

```
NOW TEST DRIVE ERROR CONDITIONS TO SEE
IF DOING A READ-IN-PRESET FUNCTION CAUSED
ANY UNEXPECTED DRIVE ERRORS.
```

```
if .GO IS_SET
then
```

```
!DID THE RD-IN-PRE FUNC HANG THE GO BIT
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (73)

```

13485 :ML4AD
13486 :
13487 :
13488 : 6543
13489 : 6544
13490 : 6545
13491 : 6546
13492 : 6547
13493 : 6548
13494 : 6549
13495 : 6550
13496 : 6551
13497 : 6552
13498 : 6553
13499 : 6554
13500 : 6555
13501 : 6556
13502 : 6557
13503 : 6558
13504 : 6559
13505 : 6560
13506 : 6561
13507 : 6562
13508 : 6563
13509 : 6564
13510 : 6565
13511 : 6566
13512 : 6567
13513 : 6568
13514 : 6569
13515 : 6570
13516 : 6571
13517 : 6572
13518 : 6573
13519 : 6574
13520 : 6575
13521 : 6576
13522 : 6577
13523 : 6578
13524 : 6579
13525 : 6580
13526 : 6581
13527 : 6582
13528 : 6583
13529 : 6584
13530 : 6585
13531 : 6586
13532 : 6587
13533 : 6588
13534 : 6589
13535 : 6590
13536 : 6591
13537 : 6592
13538 : 6593
13539 : 6594

TEST CODE SECTION

begin
ERRDF (62, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_11, WRD_19); !THE GO BIT IS HUNG
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS
end;

if .ILF IS_SET
then
begin
ERRDF (63, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_3, PHR_5, WRD_11, FNC_11, WRD_19); !DID THE RD-IN-PRE FUNC CAUSE A ILLEGAL FUNC ERROR
!THE ILF BIT IS SET
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS
end;

if .OPI IS_SET
then
begin
ERRDF (64, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_4, PHR_5, WRD_11, FNC_11, WRD_19); !DID THE RD-IN-PRE CAUSE A OPERATION INCOMPLETE ERROR
!THE OPI BIT IS SET
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR
end;

ENDSUB;
!END SCOOP LOOP

NOW WE'LL TEST THE VV BIT FOR CLEARING
AFTER POWER UP WITH BAD BATTERY BACK-UP,
ECC INITIALIZE CLEARING MEMORY TO ZEROES
AND DC LO CREATING 'MB_DIS_LO' AND CLEARING
OUT DRIVE REGISTERS.

BGNSUB;
IO_BUF = ONES;
DSA_CNT = -1;
!START SCOOP LOOP
!LOAD THE FIRST IO BUF WORD WITH ONES DATA
!RESET THE DSA COUNTER

WRITE A BACK GROUND OF ALL ONES TO
ALL OF MEMORY. THIS WILL PROVE IF
ECC INIT TOOK PLACE AFTER POWER UP
WITH BAD BATTERY BACK-UP.

do
begin
BREAK;
DSA_CNT = .DSA_CNT + 1;
WRT_TRANSFER (SIZE = -256, DST = .DSA_CNT, SRC = IO_BUF); !LOAD ALL OF MEMORY WITH ONES DATA
!LOOK FOR A CONTROL C
!UP THE DSA COUNTER
!DO A BLOCK WRITE TRANSFER
end;

!VER CZMLAD CHANGED TEST TO UNSIGNED TEST

until .DSA_CNT eqlU .LST_BLK;
!REPEAT WRITING UNTIL THE LAST BLOCK IS WRITTEN

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (73)

!CLEAR THE ERROR FLAG
!CLEAR THE MASS BUS

13541 :ML4AD
13542 :
13543 :
13544 : 6595
13545 : 6596
13546 : 6597
13547 : 6598
13548 : 6599
13549 : 6600
13550 : 6601
13551 : 6602
13552 : 6603
13553 : 6604
13554 : 6605
13555 : 6606
13556 : 6607
13557 : 6608
13558 : 6609
13559 : 6610
13560 : 6611
13561 : 6612
13562 : 6613
13563 : 6614
13564 : 6615
13565 : 6616
13566 : 6617
13567 : 6618
13568 : 6619
13569 : 6620
13570 : 6621
13571 : 6622
13572 : 6623
13573 : 6624
13574 : 6625
13575 : 6626
13576 : 6627
13577 : 6628
13578 : 6629
13579 : 6630
13580 : 6631
13581 : 6632
13582 : 6633
13583 : 6634
13584 : 6635
13585 : 6636
13586 : 6637
13587 : 6638
13588 : 6639
13589 : 6640
13590 : 6641
13591 : 6642
13592 : 6643
13593 : 6644
13594 : 6645
13595 : 6646

TEST CODE SECTION

ERR_FLG = ONE;
CLR_MBUS;

TELL THE OPERATOR THE SHUT OFF BOX
POWER AND AS THE AC POWER IS GOING
DOWN TEST 'UNS BIT' FOR SETTING.

PRINTB (ONE_FMT, PWR_OFF);

```
do
  begin
  BREAK;

  if .UNS
  then
    begin
    ERR_FLG = ZERO;
    exitloop;
    end;
  end
until .NED;
```

```
if .ERR_FLG
then
  begin
  ERRDF (151, ASYNC, DUMPER);
  PRINTB (ONE_FMT, UNS_ERR);
  end;
```

NOW TELL THE OPERATOR TO TURN
THE BOX POWER BACK ON. ONCE THE
DRIVE IS READY TEST THE VV BIT AND
SEE IF THE POWER UP WITH BAD BATTERY
BACK-UP CLEAR THE VV BIT, CLEARED
MEMORY AND CLEARED THE DRIVE REGISTERS.

PRINTB (ONE_FMT, PWR_ON);

```
do
  begin
  BREAK;
  end
until .DRY;

if .VV
then
```

!TELL THE OPERATOR TO TURN BOX POWER OFF

!SAMPLE THE UNS BIT UNTIL THE DRIVE TIMES OUT

!LOOK FOR A CONTROL C

!IS THE UNS BIT SET YET

!IT'S SET

!THE BIT IS NOT IN ERROR SO CLEAR THE ERROR FLAG

!EXIT THE LOOP

!REPEAT SAMPLING UNS UNTIL NED IS SET

!IS THE UNS BIT IN ERROR

!THE BIT IS IN ERROR

!REPORT THE ERROR TO THE OPERATOR

!TELL WHAT THE ERROR IS

!TELL THE OPERATOR TO TURN BOX POWER BACK ON

!DO NOTHING

!LOOK FOR CONTROL C

!UNTIL THE DRIVE IS READY

!DID THE POWER FAIL WITH BAD BAT BACK-UP CLEAR THE VV BIT

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (73)

```

13597 :ML4AD
13598 :
13599 :
13600 : 6647
13601 : 6648
13602 : 6649
13603 : 6650
13604 : 6651
13605 : 6652
13606 : 6653
13607 : 6654
13608 : 6655
13609 : 6656
13610 : 6657
13611 : 6658
13612 : 6659
13613 : 6660
13614 : 6661
13615 : 6662
13616 : 6663
13617 : 6664
13618 : 6665
13619 : 6666
13620 : 6667
13621 : 6668
13622 : 6669
13623 : 6670
13624 : 6671
13625 : 6672
13626 : 6673
13627 : 6674
13628 : 6675
13629 : 6676
13630 : 6677
13631 : 6678
13632 : 6679
13633 : 6680
13634 : 6681
13635 : 6682
13636 : 6683
13637 : 6684
13638 : 6685
13639 : 6686
13640 : 6687
13641 : 6688
13642 : 6689
13643 : 6690
13644 : 6691
13645 : 6692
13646 : 6693
13647 : 6694
13648 : 6695
13649 : 6696
13650 : 6697
13651 : 6698

TEST CODE SECTION

begin
ERRDF (152, ASYNC, DUMPER);
PRINTB (ONE_FMT, VV_SET);
end;

incr REG_SEL from 0 to 6 do
begin
selectone .REG_SEL of
set
[0] :
index = 0;
[1] :
index = 6;
[2] :
index = 3;
[3] :
index = 10;
[4] :
index = 13;
[5] :
index = 14;
[6] :
index = 8;
tes;

CLR_DATA = (.HI) or (.IGNORE);
RD_REG (.CLR_DATA, .REG_SEL, ERR_FLG);
if .ERR_FLG
then
begin
ERRDF (153, ASYNC, DUMPER);
PRINTB (ONE_FMT, MB_DIS_ERR);
PRINTB (FMT_16, .ML_REG[.index, REGISTER_ADD], .CLR_DATA, .RD_DATA);
exitloop;
end;

end;

WCE_CNT = ZERO;
DSA_CNT = -1;
IO_BUF = ZEROES;

do

```

```

!THE VV BIT DIDN'T GET CLEARED
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS

!SEE IF DC LO ON POWER UP CLEARED THE REGISTERS

!SELECT THIS REGISTERS INDEX VALUE

!TEST MLCS1

!TEST MLER

!TEST MLDA

!TEST MLMR

!TEST MLE1

!TEST MLE2

!TEST MLPA

!GENERATE THIS REGISTERS CLEARED DATA
!READ THE REG FOR ITS CLEARD DATA

!DID THE REGISTER GET CLEARED ON POWER UP

!THE REGISTER'S NOT CLEARED
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS
!EXIT THE LOOP

!CLEAR THE WRITE CHECK ERROR COUNTER
!RESET THE DSA COUNTER
!LOAD THE FIRST IO_BUF WORD WITH ZEROES DATA

!SEARCH ALL OF MEMORY FOR INIT'ED DATA

```


13989
13990
13991
13995 051176
13996 051176 004767 176242
13997 051202 104466
13998 051204 006000
13999 051206 103773
14000 051210 000207
14001
14002
14003
14008
14009
14010 ;

:ML4AD
:
TEST CODE SECTION
T21::
1\$: JSR PC,\$T21
TRAP 66
ROR R0
BLO 1\$
RTS PC

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

6720

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

6723 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 Vc(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (74)

```

14012 :ML4AD
14013 :
14014 :
14015 : 6724 :
14016 : 6725 :
14017 : 6726 : BGNTST;
14018 : 6727 :
14019 : 6728 :
14020 : 6729 : ++
14021 : 6730 : TEST NUMBER: TST 22
14022 : 6731 : TEST NAME: ILLEGAL FUNCTION TEST
14023 : 6732 :
14024 : 6733 : TEST DESCRIPTION:
14025 : 6734 :
14026 : 6735 : TEST THE DETECTION OF ILLEGAL
14027 : 6736 : FUNCTIONS WRITTEN TO MLCS1
14028 : 6737 : BY:
14029 : 6738 :
14030 : 6739 : WRITING ALL POSSIBLE ILLEGAL
14031 : 6740 : FUNCTIONS TO MLCS1. THEN
14032 : 6741 : TEST GO AND ERROR BITS CLEARED.
14033 : 6742 :
14034 : 6743 : --
14035 : 6744 :
14036 : 6745 : local
14037 : 6746 :     BAD_BITS,
14038 : 6747 :     BAD_FUNC;
14039 : 6748 :
14040 : 6749 : CLR_THRESHOLD;
14041 : 6750 : BAD_BITS = ZEROES;
14042 : 6751 :
14043 : 6752 : incr CNT_1 from 0 to 2 do
14044 : 6753 :     begin
14045 : 6754 :         BAD_BITS = .BAD_BITS + %'2';
14046 : 6755 :
14047 : 6756 :         incr CNT_2 from %'1' to %'71' by %'10' do
14048 : 6757 :             begin
14049 : 6758 :                 BGNSUB;
14050 : 6759 :                 CLR_MBUS;
14051 : 6760 :                 BAD_FUNC = .CNT_2 + .BAD_BITS;
14052 : 6761 :                 FIRST_BLK_XFER ?);
14053 : 6762 :                 MLCS1 = .BAD_FUNC;
14054 : 6763 :
14055 : 6764 :                 if .ILF IS_SET
14056 : 6765 :                     then
14057 : 6766 :                         begin
14058 : 6767 :
14059 : 6768 :                             if .GO IS_SET
14060 : 6769 :                                 then
14061 : 6770 :                                     begin
14062 : 6771 :                                         CMP_THRESHOLD;
14063 : 6772 :                                         ERRDF (66, ASYNC, DUMPER);
14064 : 6773 :                                         PRINTB (FIV_FMT, WRD_1, PHR_2, WRD_11, FNC_12, WRD_19);
14065 : 6774 :                                         PRINTB (FMT_12, .BAD_FUNC);
14066 : 6775 :                                     end;

```

```

!STORES A COUNT TO GENERATE BAD FUNCTIONS
!STORES GENERATED BAD FUNCTION
!CLEAR ERROR PRINT THRESHOLD
!CLEAR BAD BITS
!REPEAT LOOP 3 TIMES
!ADD 2 TO BAD_BITS
!REPEAT LOOP GENERATING 'GOOD' FUNCTIONS
!ADD BAD BITS TO CNT 2 GENERATING BAD FUNCTIONS
!SET UP X FIRST BLOCK XFERR
!LOAD MLCS1 WITH TWO BAD FUNCTIONS
!SEE IF ILF IS SET
!IF ILF IS SET THEN SEE IF GO IS SET
!COMPARE ERROR PRINT THRESHOLD
!ERROR IF GO SET WITH BAD FUNCTION

```

14068 :ML4AD
14069 :
14070 :
14071 : 6776
14072 : 6777
14073 : 6778
14074 : 6779
14075 : 6780
14076 : 6781
14077 : 6782
14078 : 6783
14079 : 6784
14080 : 6785
14081 : 6786
14082 : 6787
14083 : 6788
14084 : 6789
14085 : 6790
14086 : 6791
14087 : 6792
14088 : 6793
14089 : 6794
14090 : 6795
14091 : 6796
14092 : 6797
14093 : 6798
14094 : 6799
14095 : 6800

TEST CODE SECTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (74)

```

end
else
begin
  CMP THRESHOLD;          !COMPARE ERROR PRINT THRESHOLD
  ERRDF (67, ASYNC, DUMPER); !ERROR BAD FUNCTION DON'T CAUSE ILF
  PRINTB (FIV_FMT, WRD_3, PHR_1, WRD_1, FNC_12, WRD_19);
  PRINTB (FMT_12, .BAD_FUNC);
end;

if .OPI IS_SET          !SEE IF OPI IS SET
then
begin
  CMP THRESHOLD;          !COMPARE ERROR PRINT THRESHOLD
  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;

14100
14104 051212 004167 133374
14105 051216 005067 144154
14106 051222 005004
14107 051224 005001
14108 051226 062704 000002
14109 051232 012702 000001
14110 051236 010405
14111 051240 060205
14112 051242 104402
14113 051244 152777 000040 144172
14114 051252 016700 144554
14115 051256 042700 177770
14116 051262 142777 000007 144154
14117 051270 150077 144150
14118 051274 010503
14119 051276 004767 146760
14120 051302 010377 144076
14121 051306 132777 000001 144150
14122 051314 001446

```

.SBTTL $T22 TEST CODE SECTION
$T22: JSR R1,$$SAVE5
      CLR P.CNT
      CLR
      CLR
1$:   ADD
      MOV #1,CNT
2$:   MOV R4,R5
      ADD R2,R5
3$:   TRAP 2
      BISB #40,@ML.REG+40
      MOV ML.DUT,RO
      BIC #177770,RO
      BICB #7,@ML.REG+40
      BISB RO,@ML.REG+40
      MOV R5,R3
      JSR PC,FIRST.BLK.XFER
      MOV R3,@ML.REG
      BITB #1,@ML.REG+60
      BEQ 4$

```

```

:
:
: BAD.BITS
: CNT.1
: *.BAD.BITS
: *.CNT.2
: BAD.BITS,*
: CNT.2,*
:
:
: *.BAD.FUNC
:
: BAD.FUNC,*
:
6722
6747
6750
6752
6754
6756
6760
6757
6758
6760
6761
6762
6764

```


Address	OpCode	Operand 1	Operand 2	Comment	Label
14180					
14181					
14182					
14183	051566	000104			
14184	051570	012706			
14185	051572	026302			
14186	051574	012746	010730		
14187	051600	012746	012322		6791
14188	051604	012746	010650		
14189	051610	012746	011714		
14190	051614	012746	010544		
14191	051620	012746	010304		
14192	051624	012746	000006		
14193	051630	010600			
14194	051632	104414			
14195	051634	010316			
14196	051636	012746	007224		6792
14197	051642	012746	000002		
14198	051646	010600			
14199	051650	104414			
14200	051652	062706	000022		
14201	051656	104467			6788
14202	051660	006000			6793
14203	051662	103002			
14204	051664	000167	177352		
14205	051670	062702	000010		
14206	051674	020227	000071		6756
14207	051700	003002			
14208	051702	000167	177330		
14209	051706	005201			
14210	051710	020127	000002		6752
14211	051714	003002			
14212	051716	000167	177304		
14213	051722	000207			6722
14214					
14215					
14216					
14221					
14222					
14226					
14227					
14231	051724				
14232	051724	004767	177262		
14233	051730	104466			
14234	051732	006000			6798

:ML4AD
: TEST CODE SECTION

```

        .WORD 104
        .WORD ASYNC
        .WORD DUMPER
        MOV #WRD.19,-(SP)
        MOV #FNC.12,-(SP)
        MOV #WRD.12,-(SP)
        MOV #PHR.5,-(SP)
        MOV #WRD.4,-(SP)
        MOV #FIV.FMT,-(SP)
        MOV #6,-(SP)
        MOV SP,R0
        TRAP 14 ; SP,*
        MOV R3,(SP) ; BAD.FUNC,*
        MOV #FMT.12,-(SP)
        MOV #2,-(SP)
        MOV SP,R0 ; SP,*
        TRAP 14
        ADD #22,SP
        TRAP 67
        ROR R0
        BHIS 8$
        JMP 3$
        ADD #10,R2
        CMP R2,#71 ; *,CNT.2
        BGT 9$ ; CNT.2,*
        JMP 2$
        INC R1
        CMP R1,#2 ; CNT.1
        BGT 10$ ; CNT.1,*
        JMP 1$
        RTS PC
    
```

: Routine Size: 165 words
: Maximum stack depth per invocation: 15 words

T22:: .SBTTL T22 TEST CODE SECTION

```

1$: JSR PC,$T22
    TRAP 66
    ROR R0
    
```

14236
14237
14238
14239 051734 103773
14240 051736 000207
14241
14242
14243
14248
14249
14250 ;

:ML4AD
:

TEST CODE SECTION

BLO 1\$
RTS PC

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

6801 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (75)

14252 :ML4AD
14253 :
14254 :
14255 :
14256 :
14257 :
14258 :
14259 :
14260 :
14261 :
14262 :
14263 :
14264 :
14265 :
14266 :
14267 :
14268 :
14269 :
14270 :
14271 :
14272 :
14273 :
14274 :
14275 :
14276 :
14277 :
14278 :
14279 :
14280 :
14281 :
14282 :
14283 :
14284 :
14285 :
14286 :
14287 :
14288 :
14289 :
14290 :
14291 :
14292 :
14293 :
14294 :
14295 :
14296 :
14297 :
14298 :
14299 :
14300 :
14301 :
14302 :
14303 :
14304 :
14305 :
14306 :

6802
6803
6804
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

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 23

TEST NAME: REGISTER MODIFICATION REFUSED TEST

TEST DESCRIPTION:

TEST THE DETECTION OF A
REGISTER MODIFICATION REFUSED
BY:

1. WRITTING TO MLCS1, MLDA
AND MLER WHILE THE DRIVE
IS BUSY AND TEST RMR
BIT SET.

ALSO SEE IF THE DRIVE ASSERTED
EXCEPTION BY TESTING THE TRE BIT SET.

!--

incr CNT from 0 to 2 do

!REPEAT LOOP 3 TIMES

begin

BGNSUB;

CLR MBUS;

MLCS1 = write;

!DO A WRITE FUNCTION

case .CNT from 0 to 2 of

!WRITE TO SELECTED REGISTERS FORCING RMR

set

[0] :

MLCS1 = %0'000000';

[1] :

MLDA = ONES;

[2] :

MLER = ONES

tes;

DELAY (FRTY_US);

if .RMR IS_NOT_SET

!SEE IF RMR GOT SET

then

begin

ERRDF (69, ASYNC, DUMPER);

!ERROR IF NOT SET

PRINTB (FOR_FMT, WRD_21, PHR_1, WRD_11, WRD_21);

end;

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (75)

!SEE IF DRIVE ASSERTED EXCEPTION BY TESTING TRE

```

14308 :ML4AD
14309 :
14310 :
14311 : 6854
14312 : 6855
14313 : 6856
14314 : 6857
14315 : 6858
14316 : 6859
14317 : 6860
14318 : 6861
14319 : 6862
14320 : 6863
14321 : 6864
14322 : 6865
14326 :
14327 :
14331 : 051740 004167 132574
14332 : 051744 005746
14333 : 051746 005002
14334 : 051750 104402
14335 : 051752 152777 000040 143464
14336 : 051760 016701 144046
14337 : 051764 042701 177770
14338 : 051770 142777 000007 143446
14339 : 051776 150177 143442
14340 : 052002 012777 000061 143374
14341 : 052010 010201
14342 : 052012 006301
14343 : 052014 066107 052020
14344 : 052020 000006
14345 : 052022 000014
14346 : 052024 000024
14347 : 052026 005077 143352
14348 : 052032 000407
14349 : 052034 012777 177777 143372
14350 : 052042 000403
14351 : 052044 012777 177777 143412
14352 : 052052 012700 000050
14353 : 052056 001410
14354 : 052060 016701 130032
14355 : 052064 001403
14356 : 052066 005016
14357 : 052070 005301
14358 : 052072 001375
14359 : 052074 005300
14360 : 052076 000767
14361 : 052100 132777 000004 143356
14362 : 052106 001024
    
```

TEST CODE SECTION

```

if .TRE IS_NOT_SET
then
begin
ERRDF (117, SYNC, DUMPER);
PRINTB (FOR_FMT, WRD_20, PHR_1, WRD_11, WRD_21);
end;
ENDSUB;
end;
ENDTST;
    
```

.SBTTL \$T23 TEST CODE SECTION

```

$T23: JSR R1,$SAVE2
TST -(SP)
CLR R2
1$: TRAP 2
BISB #40,@ML.REG+40
MOV ML,DUT,R1
BIC #177770,R1
BICB #7,@ML.REG+40
BISB R1,@ML.REG+40
MOV #61,@ML.REG
MOV R2,R1
ASL R1
ADD 2$(R1),PC
2$: .WORD 3$(R1)-2$
.WORD 4$(R1)-2$
.WORD 5$(R1)-2$
3$: CLR @ML.REG
BR 6$
4$: MOV #-1,@ML.REG+30
BR 6$
5$: MOV #-1,@ML.REG+60
6$: MOV #50,R0
7$: BEQ 10$
MOV LSDLY,R1
BEQ 9$
8$: CLR (SP)
DEC R1
BNE 8$
9$: DEC R0
BR 7$
10$: BITB #4,@ML.REG+60
BNE 11$
    
```

6800
: CNT
6827
6828
6829
6831
6833
6837
6833
6840
6833
6843
6846
: *,SSTMP2
: *,SSTMP1
: SSTMP
: SSTMP1
: SSTMP2
6848

14424 052262
14425 052262 004767 177452
14426 052266 104466
14427 052270 006000
14428 052272 103773
14429 052274 000207

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

6863

14430
14431
14432
14437
14438
14439 ;

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

6866 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (76)

14441 :ML4AD
14442 :
14443 :
14444 :
14445 :
14446 :
14447 :
14448 :
14449 :
14450 :
14451 :
14452 :
14453 :
14454 :
14455 :
14456 :
14457 :
14458 :
14459 :
14460 :
14461 :
14462 :
14463 :
14464 :
14465 :
14466 :
14467 :
14468 :
14469 :
14470 :
14471 :
14472 :
14473 :
14474 :
14475 :
14476 :
14477 :
14478 :
14479 :
14480 :
14481 :
14482 :
14483 :
14484 :
14485 :
14486 :
14487 :
14488 :
14489 :
14490 :
14491 :
14492 :
14493 :
14494 :
14495 :

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 24

TEST NAME : initial PROM TEST

TEST DESCRIPTION:

TEST THE MEMORY ARRAYS' PROM
TIMING AND CONTROL LOGIC FOR
INITIAL PROM READS BY:

1. READING 14 PROM LOCATIONS
AND TESTING FOR:
 - A. CHECK SUM ERRORS AT
EACH ROW COLUMN ADRS
 - B. SUM OF EACH PROM BIT SET
<9,0> GTR 14.

Local

OFF_SET_CNT : vector [10, byte],
ROW_ORED_COL : bitvector [16],
R_C_SAV : bitvector [16],
PROM_ADRS,
CHK_SUM,
CHK_SUM_ERR,
BAD_NIB_CNT,
CNT_14_BAD,
DODD_FLG,
ERR_FLG;

!COUNTS EACH NIBBLE OFFSET
!SAVES ROW DATA OR'ED 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

CLR_THRESHOLD;
CLR_MBUS;
DODD_FLG = ZERO;
PROM_DIS = ONE;
CHK_SUM_ERR = ZEROES;

!CLEAR ERROR PRINT THRESHOLD

!SET PROM DISABLE MODE

incr CNT from 0 to 9 do
OFF_SET_CNT [.CNT] = ZEROES;

!CLEAR OFFSET COUNTS

incr ADRS_CNT from 0 to 14 do
begin
ROW_ORED_COL = ZEROES;
PROM_ADRS = .ADRS_CNT;

!READ PROM DATA FROM 15 ARRAY WORDS

!CLEAR ROW ORED COL SAVE LOCATION
!GET COPY OF ADRS_CNT

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (76)

14497 :ML4AD
14498 :
14499 :
14500 :
14501 :
14502 :
14503 :
14504 :
14505 :
14506 :
14507 :
14508 :
14509 :
14510 :
14511 :
14512 :
14513 :
14514 :
14515 :
14516 :
14517 :
14518 :
14519 :
14520 :
14521 :
14522 :
14523 :
14524 :
14525 :
14526 :
14527 :
14528 :
14529 :
14530 :
14531 :
14532 :
14533 :
14534 :
14535 :
14536 :
14537 :
14538 :
14539 :
14540 :
14541 :
14542 :
14543 :
14544 :
14545 :
14546 :
14547 :
14548 :
14549 :
14550 :
14551 :

TEST CODE SECTION

```

incr TWICE from 0 to 1 do
  begin
    BAD_NIB_CNT = ZEROES;
    ERR_FLG = ZERO;
    MLPA = .PROM_ADRS;
    DELAY (ONE_US);
    R_C_SAV = .MLPD;
    !READ ROW AND COL DATA FOR THIS ARRAY WORD

    incr BIT_CNT from 0 to 9 do
      !COUNT NUMBER BITS SET IN <9:0>
      if .R_C_SAV [.BIT_CNT] IS_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;

    CHK_SUM = .R_C_SAV<10, 3>;
    !GET THE CHECK SUM BITS

    if .R_C_SAV [13] IS_SET then CHK_SUM = .CHK_SUM + 1;
    !ADD IN BIT 13

    if .R_C_SAV [14] IS_SET then CHK_SUM = .CHK_SUM + 1;
    !ADD IN BIT 14

    if .R_C_SAV [15] IS_SET then CHK_SUM = .CHK_SUM + 1;
    !ADD IN BIT 15

    if .R_C_SAV [15] IS_SET
    then
      !IS BIT 15 SET
      begin
        if .BAD_NIB_CNT lss .CHK_SUM then ERR_FLG = ONE;
        !SET ERROR FLG IF CHECK SUM ERROR

      end
    else
      begin
        if .BAD_NIB_CNT neq .CHK_SUM then ERR_FLG = ONE;
        !SET ERROR FLG IF CHECK SUM ERROR

      end;

    if .ERR_FLG IS_SET
    then
      !WAS THERE A CHECK SUM ERROR
      begin
        !REPORT INTERMEDIATE ERROR IF YES
        !COMPARE ERROR PRINT THRESHOLD
        CMP THRESHOLD;
        ERRDF (70, INTER, DUMPER);
        PRINTB (SIX_FMT, FNC_21, WRD_10, WRD_12, WRD_45, WRD_35, FNC_6);
        PRINTB (FMT_10, .CHK_SUM, .R_C_SAV);
        DODU_FLG = ONE;

      end;

    ROW_ORED_COL = (.ROW_ORED_COL) or (.R_C_SAV);
    !OR ROW AND COLUMN DATA
    PROM_ADRS = (.PROM_ADRS) or (%'2000');
    !GET COLUMN DATA

    end;

    incr index from 0 to 9 do
      !GET TOTAL OFF SET COUNTS FROM THE 15 ARRAY WORDS
      if .ROW_ORED_COL [.index] IS_SET then OFF_SET_CNT [.index] = .OFF_SET_CNT [.index] + 1;
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (7J)

```

14553 :ML4AD
14554 :
14555 :
14556 : 6971     end;
14557 : 6972
14558 : 6973     CNT_14_BAD = ZEROES;
14559 : 6974
14560 : 6975     incr CNT from 0 to 9 do
14561 : 6976
14562 : 6977         if .OFF_SET_CNT [.CNT] geq 14 then CNT_14_BAD = .CNT_14_BAD + 1;
14563 : 6978
14564 : 6979     if .CNT_14_BAD neq ZERO
14565 : 6980     then
14566 : 6981         begin
14567 : 6982             ERRDF (71, ARR_DAT, DUMPER);
14568 : 6983
14569 : 6984         if .CNT_14_BAD eql 10
14570 : 6985         then
14571 : 6986             PRINTB (SIX_FMT, WRD_55, WRD_54, WRD_46, WRD_51, WRD_42, WRD_40)
14572 : 6987         else
14573 : 6988             begin
14574 : 6989                 PRINTB (FOR_FMT, WRD_46, WRD_47, WRD_42, WRD_40);
14575 : 6990
14576 : 6991             incr CNT from 0 to 9 do
14577 : 6992
14578 : 6993                 if .OFF_SET_CNT [.CNT] geq 14 then PRINTB (FMT_13, .CNT, (.OFF_SET_CNT [.CNT]));
14579 : 6994
14580 : 6995             end;
14581 : 6996
14582 : 6997         DODU_FLG = ONE;
14583 : 6998         end;
14584 : 6999
14585 : 7000     if .DODU_FLG IS_SET
14586 : 7001     then
14587 : 7002         begin
14588 : 7003             DODU (.ML_LUN);
14589 : 7004             DOCLN;
14590 : 7005         end;
14591 : 7006
14592 : 7007     ENDTST;
14593 : 7008
14597 :
14598 :

```

```

!COUNT HOW MANY OFFSET COUNTS GEQ 14
!WAS ANY OFF_SET COUNTS GTR ZERO
!ERROR IF YES
!WERE ALL 10 NIBBLE OFFSETS GEQ 14
!ERROR IF YES
!IF NO THEN PRINT WHICH ONES WERE
!PRINT MESSAGE
!FIND OFFSETS GEQ 14
!PRINT NIBBLE POSITION AND COUNT
!DROP THIS UNIT IF DODU_FLG SET

```

```

14602 052276 004167 132310
14603 052302 162706 000030
14604 052306 005067 143064
14605 052312 152777 000040 143124
14606 052320 016705 143506
14607 052324 042705 177770

.SBTTL $T24 TEST CODE SECTION
$T24: JSR R1,$SAVE5
SUB #30,SP
CLR P,CNT
BISB #40,ML.REG+40
MOV ML.DUT,R5
BIC #177770,R5

```

6865
6903
6905

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Line No.
14721							
14722							
14723							
14724	053032	020427	000001		CMP R4,#1	: TWICE,*	
14725	053036	003002			BGT 17\$		
14726	053040	000167	177350		JMP 3\$		
14727	053044	005002		17\$:	CLR R2	: INDEX	
14728	053046	010201		18\$:	MOV R2,R1	: INDEX,*	6967
14729	053050	006201			ASR R1		6969
14730	053052	006201			ASR R1		
14731	053054	006201			ASR R1		
14732	053056	012700	000010		MOV #10,R0		
14733	053062	060600			ADD SP,R0	: ROW.ORED.COL,*	
14734	053064	060001			ADD R0,R1		
14735	053066	010146			MOV R1,-(SP)		
14736	053070	010246			MOV R2,-(SP)	: INDEX,*	
14737	053072	042716	177770		BIC #177770,(SP)		
14738	053076	012746	000001		MOV #1,-(SP)		
14739	053102	005046			CLR -(SP)		
14740	053104	004767	130524		JSR PC,BLSGT2		
14741	053110	062706	000010		ADD #10,SP		
14742	053114	005300			DEC R0		
14743	053116	001005			BNE 19\$		
14744	053120	012701	000016		MOV #16,R1		
14745	053124	060601			ADD SP,R1	: OFF.SET.CNT,*	
14746	053126	060201			ADD R2,R1	: INDEX,*	
14747	053130	105211			INCB (R1)		
14748	053132	005202		19\$:	INC R2	: INDEX	
14749	053134	020227	000011		CMP R2,#11	: INDEX,*	6967
14750	053140	003742			BLE 18\$		
14751	053142	005205			INC R5	: ADRS.CNT	
14752	053144	020527	000016		CMP R5,#16	: ADRS.CNT,*	6914
14753	053150	003002			BGT 20\$		
14754	053152	000167	177224		JMP 2\$		
14755	053156	005000		20\$:	CLR R0	: CNT.14.BAD	
14756	053160	005001			CLR R1	: CNT	6973
14757	053162	012702	000016	21\$:	MOV #16,R2	: CNT	6975
14758	053166	060602			ADD SP,R2	: OFF.SET.CNT,*	6977
14759	053170	060102			ADD R1,R2	: CNT,*	
14760	053172	121227	000016		CMPB (R2),#16		
14761	053176	103401			BLO 22\$		
14762	053200	005200			INC R0	: CNT.14.BAD	
14763	053202	005201		22\$:	INC R1	: CNT	
14764	053204	020127	000011		CMP R1,#11	: CNT,*	6975
14765	053210	003764			BLE 21\$		
14766	053212	005700			TST R0	: CNT.14.BAD	6979
14767	053214	001505			BEQ 27\$		
14768	053216	104455			TRAP 55	:	6982
14769	053220	000107			.WORD 107		
14770	053222	013012			.WORD ARR.DAT		
14771	053224	026302			.WORD DUMPER		
14772	053226	020027	000012		CMP R0,#12	: CNT.14.BAD,*	
14773	053232	001024			BNE 23\$		6984
14774	053234	012746	011150		MOV #WRD.40,-(SP)		
14775	053240	012746	011170		MOV #WRD.42,-(SP)		6986

14838
14839
14843 053456
14844 053456 004767 176614
14845 053462 104466
14846 053464 006000
14847 053466 103773
14848 053470 000207
14849
14850
14851
14856
14857
14858 ; 7009 !<BLF/PAGE>

.SBTTL T24 TEST CODE SECTION
T24::
1\$: JSR PC,\$T24
TRAP 66
ROR R0
BLO 1\$
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

7006

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (77)

14860 :ML4AD
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 :
14896 :
14897 :
14898 :
14899 :
14900 :
14901 :
14902 :
14903 :
14904 :
14905 :
14906 :
14907 :
14908 :
14909 :
14910 :
14911 :
14912 :
14913 :
14914 :

TEST CODE SECTION

```

7010 :
7011 :
7012 : BGNTST;
7013 :
7014 : !++
7015 : TEST NUMBER: TST 25
7016 :
7017 : TEST NAME: PROM OR FUNCTION TEST
7018 :
7019 : TEST DESCRIPTION:
7020 :
7021 : TEST THE HARDWARE ORING OF
7022 : THE MEMORY ARRAYS' PROM
7023 : ROW COLUMN DATA BY:
7024 :
7025 : 1. READING AND STORING 128
7026 : HARDWARE ORED ROW COLUMN
7027 : DATA.
7028 :
7029 : 2. THEN IN PROM DISABLE MODE
7030 : AND VIA SOFTWARE CONTROL,
7031 : READ AND OR PROM ROW
7032 : COLUMN DATA AND COMPARE
7033 : AGAINST THE RESPECTIVE
7034 : STORED HARDWARE ORED DATA.
7035 :
7036 :
7037 :
7038 : local
7039 : R_BITS,
7040 : C_BITS,
7041 : SW_ORED,
7042 : HW_SAVE,
7043 : DODU_FLG;
7044 :
7045 : BGNSUB;
7046 : CLR_THRESHOLD;
7047 : CLR_MBUS;
7048 : DODU_FLG = ZERO;
7049 : DAT_DM = ONE;
7050 : FIRST_BLK_XFER ();
7051 : ML_FUNC = write;
7052 :
7053 : incr PROM_ADRS from 0 to 127 do
7054 : begin
7055 : DELAY (ONE US);
7056 : HW_OR_TBL [PROM_ADRS] = .MLPD;
7057 : DAT_CLK = ONE;
7058 : end;
7059 :
7060 : CLR_MBUS;
7061 : PROM_DIS = ONE;
    
```

```

!PROM ROW DATA
!PROM COL DATA
!SOFTWARE CALCULATED PROM ORED DATA
!SOFTWARE PROM ORED DATA
!DROP UNIT FLAG

!CLEAR ERROR PRINT THRESHOLD

!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
    
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (77)

```

14916 ;ML4AD
14917 ;
14918 ;
14919 : 7062
14920 : 7063   incr PROM_ADRS from 0 to 127 do
14921 : 7064   begin
14922 : 7065   MLPA = .PROM_ADRS;
14923 : 7066   DELAY (ONE US);
14924 : 7067   R BITS = .MLPD;
14925 : 7068   MLPA = .PROM_ADRS or %o'2000';
14926 : 7069   DELAY (ONE US);
14927 : 7070   C BITS = .MLPD;
14928 : 7071   SW_ORED = .R BITS or .C BITS;
14929 : 7072   HW_SAVE = .HW_OR_TBL [.PROM_ADRS];
14930 : 7073
14931 : 7074   if (.HW_SAVE<0, 9>) neq (.SW_ORED<0, 9>)
14932 : 7075   then
14933 : 7076   begin
14934 : 7077   CMP THRESHOLD;
14935 : 7078   ERRDF (76, ARR DAT, DUMPER);
14936 : 7079   PRINTB (FOR_FMT, WRD 35, WRD 36, WRD 19, PHR 4);
14937 : 7080   PRINTR (FMT_2, .SW_ORED<0, 9>, .HW_SAVE<0, 9>, (.SW_ORED<0, 9> xor .HW_SAVE<0, 9>));
14938 : 7081   DODU_FLG = ONE;
14939 : 7082   end;
14940 : 7083
14941 : 7084   end;
14942 : 7085
14943 : 7086   ENDSUB;
14944 : 7087
14945 : 7088   if .DODU_FLG IS_SET
14946 : 7089   then
14947 : 7090   begin
14948 : 7091   DODU (.ML_LUN);
14949 : 7092   DOCLN;
14950 : 7093   end;
14951 : 7094
14952 : 7095   ENDTST;
14956
14957

```

```

!CALCULATE 128 SW ORED DATA & COMPARE TO HW TABLE
!LOADING MLPA INITIATES A PROM READ
!SAVE ROW DATA
!ENABLE COLUMN DATA ADRS
!SAVE COL DATA
!CALCULATE SOFTWARE ORED
!GET RESPECTIVE HARDWARE ORED
!COMPARE SW & HW ORED
!IF NEQ THEN ERROR
!COMPARE ERROR PRINT THRESHOLD
!DROP THIS UNIT IF DODU FLG IS_SET

```

```

14961 053472 004167 131114   $T25:   .SBTTL   $T25 TEST CODE SECTION
14962 053476 162706 000010   JSR     R1,$SAVE5
14963 053502 104402           SUB     #10,SP
14964 053504 005067 141666   1$:    TRAP   2
14965 053510 152777 000040 141726 CLR     P.CNT
14966 053516 016705 142310   BISB   #40,@ML.REG+40
14967 053522 042705 177770   MOV    ML.DUT,R5
14968 053526 142777 000007 141710 BIC    #177770,R5
14969 053534 150577 141704   BICB   #7,@ML.REG+40
14970 053540 005016           BISB   R5,@ML.REG+40
                           CLR     (SP)
                           ; DODU.FLG

```

```

7008
7043
7045
7046
7048

```


Address	OpCode	Operand1	Operand2	Operand3	Comment	Time	Page
15028							
15029							
15030							
15031	054024	016601	000004		MOV 4(SP),R1	29-Mar-1982 16:23:04	TOPS
15032	054030	056601	000002		BIS 2(SP),R1	29-Mar-1982 16:21:03	PA:<
15033	054034	010305			MOV R3,R5		7071
15034	054036	006305			ASL R5		7072
15035	054040	016502	013402		MOV HW.OR.TBL(R5),R2		
15036	054044	010104			MOV R1,R4		
15037	054046	042704	177000		BIC #177000,R4		7074
15038	054052	010205			MOV R2,R5		
15039	054054	042705	177000		BIC #177000,R5		
15040	054060	020504			CMP R5,R4		
15041	054062	001457			BEQ 16\$		
15042	054064	005267	141306		INC P.CNT		
15043	054070	026767	141302	141302	CMP P.CNT,LIMIT		7076
15044	054076	003055			BGT 17\$		
15045	054100	104455			TRAP 55		
15046	054102	000114			.WORD 114		7078
15047	054104	013012			.WORD ARR.DAT		
15048	054106	026302			.WORD DUMPER		
15049	054110	012746	011676		MOV #PHR.4,-(SP)		
15050	054114	012746	010730		MOV #WRD.19,-(SP)		7079
15051	054120	012746	011116		MOV #WRD.36,-(SP)		
15052	054124	012746	011110		MOV #WRD.35,-(SP)		
15053	054130	012746	010270		MOV #FOR.FMT,-(SP)		
15054	054134	012746	000005		MOV #5,-(SP)		
15055	054140	010600			MOV SP,R0		
15056	054142	104414			TRAP 14		: SP,*
15057	054144	010205			MOV R2,R5		
15058	054146	010116			MOV R1,(SP)		: HW.SAVE,*
15059	054150	010504			MOV R5,R4		: SW.ORED,*
15060	054152	040104			BIC R1,R4		
15061	054154	040516			BIC R5,(SP)		
15062	054156	050416			BIS R4,(SP)		
15063	054160	010246			MOV R2,-(SP)		: HW.SAVE,*
15064	054162	042716	177000		BIC #177000,(SP)		
15065	054166	010146			MOV R1,-(SP)		: SW.ORED,*
15066	054170	042716	177000		BIC #177000,(SP)		
15067	054174	012746	006506		MOV #FMT.2,-(SP)		
15068	054200	012746	000004		MOV #4,-(SP)		
15069	054204	010600			MOV SP,R0		: SP,*
15070	054206	104414			TRAP 14		
15071	054210	012766	000001	000024	MOV #1,24(SP)		: *,DODU.FLG
15072	054216	062706	000024		ADD #24,SP		7081
15073	054222	005203			INC R3		7076
15074	054224	020327	000177	16\$:	CMP R3,#177		7063
15075	054230	003630			BLE 7\$		
15076	054232	104467			TRAP 67		
15077	054234	006000			ROR R0		7084
15078	054236	103002			BHIS 18\$		
15079	054240	000167	177236		JMP 1\$		
15080	054244	021627	000001	18\$:	CMP (SP),#1		: DODU.FLG,*
15081	054250	001004			BNE 19\$		7088
15082	054252	016700	141552		MOV ML.LUN,R0		7091

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

15084
 15085
 15086
 15087 054256 104451
 15088 054260 104444
 15089 054262 062706 000010
 15090 054266 000207
 15091
 15092
 15093
 15098
 15099
 15103
 15104
 15108 054270
 15109 054270 004767 177176
 15110 054274 104466
 15111 054276 006000
 15112 054300 103773
 15113 054302 000207
 15114
 15115
 15116
 15121
 15122
 15123 ; 7096 !<BLF/PAGE>

;ML4AD
;
TEST CODE SECTION

TRAP 51
 TRAP 44
 19\$: ADD #10,SP
 RTS PC ;

7008

; Routine Size: 191 words
; Maximum stack depth per invocation: 20 words

.SBTTL T25 TEST CODE SECTION

T25::
 1\$: JSR PC,ST25
 TRAP 66 ;
 ROR R0
 BLO 1\$
 RTS PC

7093

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (78)

15125 :ML4AD
15126 :
15127 :
15128 :
15129 :
15130 :
15131 :
15132 :
15133 :
15134 :
15135 :
15136 :
15137 :
15138 :
15139 :
15140 :
15141 :
15142 :
15143 :
15144 :
15145 :
15146 :
15147 :
15148 :
15149 :
15150 :
15151 :
15152 :
15153 :
15154 :
15155 :
15156 :
15157 :
15158 :
15159 :
15160 :
15161 :
15162 :
15163 :
15164 :
15165 :
15166 :
15167 :
15168 :
15169 :
15170 :
15171 :
15172 :
15173 :
15174 :
15175 :
15176 :
15177 :
15178 :
15179 :

7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122
7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148

TEST CODE SECTION

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 '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

HIGH_CNT,
PROM_DATA : bitvector [16],
LOW_CNT,
TEMP,
ERR_FLG,
GTR_FLG;

CLR_THRESHOLD;
PROP_DATA = -1;

do

begin
PROM_DATA = .PROM_DATA + 1;
BGNSDB;
CLR_MBUS;
ERR_FLG = ZERO;
LOW_CNT = ZEROES;
HIGH_CNT = ZEROES;
GTR_FLG = ZERO;

!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

!CLEAR ERROR PRINT THRESHOLD

!TEST ALL POSSIBLE PROM DATA COMBINATIONS

!INCREMENT PROM_DATA

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (78)

```

15181 :ML4AD
15182 :
15183 :
15184 : 7149
15185 : 7150
15186 : 7151
15187 : 7152
15188 : 7153
15189 : 7154
15190 : 7155
15191 : 7156
15192 : 7157
15193 : 7158
15194 : 7159
15195 : 7160
15196 : 7161
15197 : 7162
15198 : 7163
15199 : 7164
15200 : 7165
15201 : 7166
15202 : 7167
15203 : 7168
15204 : 7169
15205 : 7170
15206 : 7171
15207 : 7172
15208 : 7173
15209 : 7174
15210 : 7175
15211 : 7176
15212 : 7177
15213 : 7178
15214 : 7179
15215 : 7180
15216 : 7181
15217 : 7182
15218 : 7183
15219 : 7184
15220 : 7185
15221 : 7186
15222 : 7187
15223 : 7188
15224 : 7189
15225 : 7190
15226 : 7191
15227 : 7192
15228 : 7193
15229 : 7194
15230 : 7195
15231 : 7196
15232 : 7197
15233 : 7198
15234 : 7199
15235 : 7200

TEST CODE SECTION

incr BIT_CNT from 0 to 9 do
    !COUNT FROM DATA BITS <9:0>
    if .PROM_DATA [.BIT_CNT] IS_SET then LOW_CNT = .LOW_CNT + 1;
HIGH_CNT = .PROM_DATA<10, 3>;
    !GET PROM DATA CHECK SUM BITS
if .PROM_DATA [13] IS_SET then HIGH_CNT = .HIGH_CNT + 1;
    !ADD IN BIT 13 IF SET
if .PROM_DATA [14] IS_SET then HIGH_CNT = .HIGH_CNT + 1;
    !ADD IN BIT 14 IF SET
if .PROM_DATA [15] IS_SET then HIGH_CNT = .HIGH_CNT + 1;
    !ADD IN BIT 15 IF SET
if .PROM_DATA [15] IS_SET then GTR_FLG = ONE;
    !SET FLAG IF BIT 15 IS SET

DAT DM XFER ();
PROM_RD = ONE;
MLPD = .PROM_DATA;
MLCS1 = write;
DAT_CLK = ONE;
!SET UP A DATA DIAG MODE XFERR
!SET PROM READ WRITE
!LOAD MLPD WITH PROM_DATA
!DO A WRITE FUNCTION
!CLOCK PROM DATA INTO UV PROM

if .UNS IS_SET
then
begin
    !SEE IF PROM DATA CAUSED A UV ERROR
    if .GTR_FLG IS_SET
    then
begin
        !UNS IS_SET. SEE IF GTR FLG IS SET
        TEMP = .PROM_DATA;
        TEMP = .TEMP and %o'162000';
        !LOAD TEMP WITH PROM DATA
        !SEE IF THESE BITS ARE SET IN PROM DATA
        if .TEMP eql %o'162000'
        then
begin
            !THESE BITS SET AUTOMATICALLY CAUSE A UNS
            if .LOW_CNT geq .HIGH_CNT
            then
begin
                !LOW<9:0> SHOULD BE ISS THAN THE HIGH<15:10> IF GTR FLG IS SET
                !ERROR IF LOW<9:0> IS GEQ HIGH<15:10>
                CMP THRESHOLD;
                !COMPARE ERROR PRINT THRESHOLD
                ERRDF (72, ARR DAT, DUMPER);
                PRINTB (SIX FMT, WRD_34, PHR_5, WRD_32, WRD_6, WRD_33, WRD_24);
                ERR_FLG = ONE;
            end;
        end;
    end;
end
else
begin
    !GTR FLG IS NOT SET
    if .LOW_CNT eql .HIGH_CNT
    then
        !LOW<9:0> SHOULD BE NEQ HIGH<15:10>

```


29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (78)

```

15293 :ML4AD
15294 :
15295 :
15296 :       7253           end
15297 :       7254
15298 :       7255           end
15299 :       7256
15300 :       7257           end;
15301 :       7258
15302 :       7259           if .ERR_FLG IS_SET
15303 :       7260           then
15304 :       7261               begin
15305 :       7262                   PRINTB (FMT_7, .PROM_DATA);
15306 :       7263                   end;
15307 :       7264
15308 :       7265           ENDSUB;
15309 :       7266           end
15310 :       7267           until .PROM_DATA eql %o'177777':
15311 :       7268
15312 :       7269           ENDTST;
15316 :
15317 :

```

```

!SEE IF ERROR FLG GOT SET
!PRINT FAILING PROM_DATA AND SET DODU_FLG
!TRY ALL BIT COMBINATIONS

```

15321	054304	004167	130302		ST26:	.SBTTL	\$T26 TEST CODE SECTION		
15322	054310	162706	000010			JSR	R1,\$SAVE5	:	7095
15323	054314	005067	141056			SUB	#10,SP	:	
15324	054320	012766	177777	000006		CLR	P,CNT	:	7134
15325	054326	005266	000006		1\$:	MOV	#-1,6(SP)	:	7137
15326	054332	104402			2\$:	INC	6(SP)	:	7141
15327	054334	152777	000040	141102		TRAP	2	:	
15328	054342	016705	141464			BISB	#40,@ML.REG+40	:	7142
15329	054346	042705	177770			MOV	ML.DUT,R5	:	
15330	054352	142777	000007	141064		BIC	#177770,R5	:	
15331	054360	150577	141060			BICB	#7,@ML.REG+40	:	
15332	054364	005001				BISB	R5,@ML.REG+40	:	
15333	054366	005066	000002			CLR	R1	:	7144
15334	054372	005002				CLR	2(SP)	:	7145
15335	054374	005066	000004			CLR	R2	:	7146
15336	054400	005005				CLR	4(SP)	:	7147
15337	054402	010504			3\$:	CLR	R5	:	7149
15338	054404	006204				MOV	R5,R4	:	7151
15339	054406	006204				ASR	R4	:	
15340	054410	006204				ASR	R4	:	
15341	054412	012703	000006			ASR	R4	:	
15342	054416	060603				MOV	#6,R3	:	
15343	054420	060304				ADD	SP,R3	:	PROM.DATA,*
15344	054422	010446				ADD	R3,R4	:	
15345	054424	010546				MOV	R4,-(SP)	:	
15346	054426	042716	177770			MOV	R5,-(SP)	:	BIT.CNT,*
15347	054432	012746	000001			BIC	#177770,(SP)	:	
						MOV	#1,-(SP)	:	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA: <

```
15517      :ML4AD
15518      :
15519      : TEST CODE SECTION
15520 05432 104467      20$: TRAP 67
15521 055434 006000      ROR R0
15522 055436 103002      BHIS 21$
15523 055440 000167 176666      JMP 2$
15524 055444 026627 000006 177777 21$: CMP 6(SP),#-1 : PROM.DATA,*
15525 055452 001402      BEQ 22$
15526 055454 000167 176646      JMP 1$
15527 055460 062706 000010      22$: ADD #10,SP
15528 055464 000207      RTS PC
15529
15530      : Routine Size: 313 words
15531      : Maximum stack depth per invocation: 18 words
15536
15537
15541
15542
15546 055466      .SBTTL T26 TEST CODE SECTION
15547 055466 004767 176612      T26::
15548 055472 104466      1$: JSR PC,$T26
15549 055474 006000      TRAP 66
15550 055476 103773      ROR R0
15551 055500 000207      BLO 1$
15552      RTS PC
15553
15554      : Routine Size: 6 words
15559      : Maximum stack depth per invocation: 0 words
15560
15561 : 7270 !<BLF/PAGE>
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (79)

15563 :ML4AD
15564 :
15565 :
15566 :
15567 :
15568 :
15569 :
15570 :
15571 :
15572 :
15573 :
15574 :
15575 :
15576 :
15577 :
15578 :
15579 :
15580 :
15581 :
15582 :
15583 :
15584 :
15585 :
15586 :
15587 :
15588 :
15589 :
15590 :
15591 :
15592 :
15593 :
15594 :
15595 :
15596 :
15597 :
15598 :
15599 :
15600 :
15601 :
15602 :
15603 :
15604 :
15605 :
15606 :
15607 :
15608 :
15609 :
15610 :
15611 :
15612 :
15613 :
15614 :
15615 :
15616 :
15617 :

TEST CODE SECTION

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

```

15619 :ML4AD
15620 :
15621 :
15622 : 7323
15623 : 7324
15624 : 7325
15625 : 7326
15626 : 7327
15627 : 7328
15628 : 7329
15629 : 7330
15630 : 7331
15631 : 7332
15632 : 7333
15633 : 7334
15634 : 7335
15635 : 7336
15636 : 7337
15637 : 7338
15638 : 7339
15639 : 7340
15640 : 7341
15641 : 7342
15642 : 7343
15643 : 7344
15644 : 7345
15645 : 7346
15646 : 7347
15647 : 7348
15648 : 7349
15649 : 7350
15650 : 7351
15651 : 7352
15652 : 7353
15653 : 7354
15654 : 7355
15655 : 7356
15656 : 7357
15657 : 7358
15658 : 7359
15659 : 7360
15660 : 7361
15661 : 7362
15662 : 7363
15663 : 7364
15664 : 7365
15665 : 7366
15666 : 7367
15667 : 7368
15668 : 7369
15672 :
15673 :

```

```

TEST CODE SECTION

DAT_DM = ONE;
FIRST_BLK_XFER ();
MLCS1 = write;

incr CNT from 0 to 4 do
begin
DELAY (ONE_US);
DAT_CLK = ONE;
end;

CLR_MBUS;
DAT_DM = ONE;
FIRST_BLK_XFER ();
MLCS1 = read;
DELAY (ONE_US);

incr ARR_WRD from 0 to 4 do
begin
DAT_CLK = ONE;
DELAY (ONE_US);
RD_LNG_WRD;

incr NIB_PTR from 0 to 9 do
begin
TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG); !COMPARE TST PAT TO NIBBLE UNDER TEST
if .ERR_FLG IS_SET then BAD_NIB_CNT = .BAD_NIB_CNT + 1;

end;

end;

TST_PAT = not .TST_PAT;
ENDSUB;
end;

if .BAD_NIB_CNT gtr 36
then
begin
ERRDF (77, ASYNC, DUMPER);
PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, WRD_45, FNC_14);
DODU (.ML_LDN);
DOCLN;
end;

ENDTST;

.SBTTL $T27 TEST CODE SECTION

```

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (79)

!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

!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	Page
15675								
15676								
15677								
15681	055502	004167	127104		ST27:	JSR R1,\$SAVE5		
15682	055506	024646				CMP -(SP),-(SP)		7269
15683	055510	012701	177777			MOV #-1,R1	*.TST.PAT	7313
15684	055514	005046				CLR -(SP)	BAD.NIB.CNT	7314
15685	055516	005005				CLR R5	TWICE	7316
15686	055520	104402			1\$:	TRAP 2		7317
15687	055522	152777	000040	137714		BISB #40,@ML.REG+40		7318
15688	055530	016704	140276			MOV ML,DUT,R4		
15689	055534	042704	177770			BIC #177770,R4		
15690	055540	142777	000007	137676		BICB #7,@ML.REG+40		
15691	055546	150477	137672			BISB R4,@ML.REG+40		
15692	055552	010177	140016			MOV R1,@ML.REG+170	TST.PAT,*	7320
15693	055556	010177	140022			MOV R1,@ML.REG+200	TST.PAT,*	7321
15694	055562	010177	137776			MOV R1,@ML.REG+160	TST.PAT,*	7322
15695	055566	152777	000010	137730		BISB #10,@ML.REG+120		7323
15696	055574	004767	142462			JSR PC,FIRST.BLK.XFER		7324
15697	055600	012777	000061	137576		MOV #61,@ML.REG		7325
15698	055606	005002				CLR R2	CNT	7327
15699	055610	012703	000001		2\$:	MOV #1,R3	*,\$STMP2	7329
15700	055614	001411			3\$:	BEQ 6\$		
15701	055616	016704	124274			MOV LSDLY,R4	*,\$STMP1	
15702	055622	001404				BEQ 5\$		
15703	055624	005066	000004		4\$:	CLR 4(SP)	\$STMP	
15704	055630	005304				DEC R4	\$STMP1	
15705	055632	001374				BNE 4\$		
15706	055634	005303			5\$:	DEC R3	\$STMP2	
15707	055636	000766				BR 3\$		
15708	055640	152777	000020	137656	6\$:	BISB #20,@ML.REG+120		7330
15709	055646	005202				INC R2	CNT	7327
15710	055650	020227	000004			CMP R2,#4	CNT,*	
15711	055654	003755				BLE 2\$		
15712	055656	152777	000040	137560		BISB #40,@ML.REG+40		7331
15713	055664	016704	140142			MOV ML,DUT,R4		
15714	055670	042704	177770			BIC #177770,R4		
15715	055674	142777	000007	137542		BICB #7,@ML.REG+40		
15716	055702	150477	137536			BISB R4,@ML.REG+40		
15717	055706	152777	000010	137610		BISB #10,@ML.REG+120		
15718	055714	004767	142342			JSR PC,FIRST.BLK.XFER		7334
15719	055720	012777	000071	137456		MOV #71,@ML.REG		7335
15720	055726	012703	000001			MOV #1,R3	*,\$STMP2	7336
15721	055732	001411			7\$:	BEQ 10\$		7337
15722	055734	016704	124156			MOV LSDLY,R4	*,\$STMP1	
15723	055740	001404				BEQ 9\$		
15724	055742	005066	000004		8\$:	CLR 4(SP)	\$STMP	
15725	055746	005304				DEC R4	\$STMP1	
15726	055750	001374				BNE 8\$		
15727	055752	005303			9\$:	DEC R3	\$STMP2	
15728	055754	000766				BR 7\$		
15729	055756	005002			10\$:	CLR R2	ARR.WRD	7339

15787
15788
15789
15790 056222 016700 137602
15791 056226 104451
15792 056230 104444
15793 056232 062706 000016
15794 056236 062706 000006
15795 056242 000207
15796
15797
15798
15803
15804
15808
15809
15813 056244
15814 056244 004767 177232
15815 056250 104466
15816 056252 006000
15817 056254 103773
15818 056256 000207
15819
15820
15821
15826
15827
15828 ; 7370 !<BLF/PAGE>

;ML4AD
;
TEST CODE SECTION
MOV ML.LUN,RO
TRAP 51
TRAP 44
ADD #16,SP
20s: ADD #6,SP
RTS PC
: Routine Size: 177 words
: Maximum stack depth per invocation: 16 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

7365

7362
7269

.SBTTL T27 TEST CODE SECTION
T27::
1s: JSR PC,ST27
TRAP 66
ROR R0
BLO 1s
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

7367

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (80)

15830 :ML4AD
15831 :
15832 :
15833 :
15834 :
15835 :
15836 :
15837 :
15838 :
15839 :
15840 :
15841 :
15842 :
15843 :
15844 :
15845 :
15846 :
15847 :
15848 :
15849 :
15850 :
15851 :
15852 :
15853 :
15854 :
15855 :
15856 :
15857 :
15858 :
15859 :
15860 :
15861 :
15862 :
15863 :
15864 :
15865 :
15866 :
15867 :
15868 :
15869 :
15870 :
15871 :
15872 :
15873 :
15874 :
15875 :
15876 :
15877 :
15878 :
15879 :
15880 :
15881 :
15882 :
15883 :
15884 :

7371
7372
7373
7374
7375
7376
7377
7378
7379
7380
7381
7382
7383
7384
7385
7386
7387
7388
7389
7390
7391
7392
7393
7394
7395
7396
7397
7398
7399
7400
7401
7402
7403
7404
7405
7406
7407
7408
7409
7410
7411
7412
7413
7414
7415
7416
7417
7418
7419
7420
7421
7422

TEST CODE SECTION

! BGNTST;

!++

TEST NUMBER: TST 28

TEST NAME: PROM SELECTION TEST

TEST DESCRIPTION:

DUE TO THE NATURE OF THE DEVICE
AND OF THE ARRAY MODULES' UV
PROMS, ONLY PROM READS ARE
ALLOWED DURING DIAG TESTING.

THEREFORE THE ARRAY MODULE UV PROMS
ARE TESTED FOR UNIQUE SELECTION BY:

1. AT EACH PRESENT ARRAY MODULE WRITE 127 ARRAY WORDS WITH 1'S/0'S PATTERN.
2. READ THE UV PROMS AT THEIR RESPECTIVE ARRAY WORD LOCATION AND SEE IF THE PROMS MASK BAD NIBBLE LOCATIONS (ENCOUNTERED BAD NIBBLES INDICATES INCORRECT MASKING). COUNT EACH BAD NIBBLE ENCOUNTERED AT AN ARRAY MODULE.
3. ALLOW A THRESHOLD OF 5 BAD NIBBLES AT ANY ARRAY MODULE.
4. REPORT PROM SEL ERRORS AT RESPECTIVE ARRAY MODULE IF THE THRESHOLD IS EXCEEDED.

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,
ERR_FLG,
TST_PAT,
ERR_CNT;

!DROP UNIT FLG
!ERROR FLG
!TEST PATTERN
!ERROR COUNT

DODU_FLG = ZERO;
TST_PAT = ONES;

!

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (80)

15942 :ML4AD
15943 :
15944 :
15945 :
15946 :
15947 :
15948 :
15949 :
15950 :
15951 :
15952 :
15953 :
15954 :
15955 :
15956 :
15957 :
15958 :
15959 :
15960 :
15961 :
15962 :
15963 :
15964 :
15965 :
15966 :
15967 :
15968 :
15969 :
15970 :
15974 :
15975 :

TEST CODE SECTION

```

end;
TST_PAT = not .TST_PAT;
end;
!REPEAT WITH COMPLIMENT DATA

if .ERR_CNT gtr 5
then
begin
ERRDF (78, ARR DAT, DUMPER);
PRINTB (THR_FMT, WRD_35, WRD_37, WRD_10);
PRINTB (FMT_9, .ARR_SEL);
DODU_FLG = ONE;
end;
!ALLOW 5 ERROR BEFORE ERRORING
!ERROR IF GTR 5

ENDSUB;

if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;
!DROP THIS UNIT IF DODU_FLG IS SET

end;
ENDTST;

```

15979 056260 004167 126326
15980 056264 162706 000016
15981 056270 005066 000010
15982 056274 012702 177777
15983 056300 016766 135514 000006
15984 056306 016766 135472 000004
15985 056314 005001
15986 056316 000167 000666
15987 056322 005066 000002
15988 056326 104402
15989 056330 005016
15990 056332 132777 000040 137104
15991 056340 016705 137466
15992 056344 042705 177770
15993 056350 142777 000007 137066
15994 056356 150577 137062
15995 056362 010277 137206
15996 056366 010277 137212

```

.SBTTL $T28: TEST CODE SECTION
JSR R1,$SAVE5
SUB #16,SP
CLR 10(SP)
MOV #-1,R2
MOV LST.ARR,6(SP)
MOV ARR.INC,4(SP)
CLR R1
JMP 25$
1$: CLR 2(SP)
2$: TRAP 2
3$: CLR (SP)
BISB #40,@ML.REG+40
MOV ML.DUT,R5
BIC #177770,R5
BICB #7,@ML.REG+40
BISB R5,@ML.REG+40
MOV R2,@ML.REG+170
MOV R2,@ML.REG+200

```

```

:
: DODU.FLG
: *.TST.PAT
:
: ARR.SEL
: ERR.CNT
: TWICE
:
: TST.PAT,*
: TST.PAT,*

```

7369
7419
7420
7425
7427
7430
7431
7433
7434

		:ML4AD				TEST CODE SECTION		29-Mar-1982 16:23:04	TOPS
		:						29-Mar-1982 16:21:03	PA:<
15998									
15999									
16000									
16001	056372	010277	137166			MOV	R2,@ML.REG+160	:	TST.PAT,*
16002	056376	152777	000010	137120		BISB	#10,@ML.REG+120	:	7435
16003	056404	012777	177400	137002		MOV	#-400,@ML.REG+10	:	7436
16004	056412	012777	014022	137004		MOV	#10.BUF,@ML.REG+20	:	7437
16005	056420	010177	137010			MOV	R1,@ML.REG+30	:	7438
16006	056424	012777	000061	136752		MOV	#61,@ML.REG	:	7439
16007	056432	005003				CLR	R3	:	7440
16008	056434	012704	000001		4\$:	MOV	#1,R4	:	CNT
16009	056440	001411			5\$:	BEQ	8\$:	*,SSTMP2
16010	056442	016705	123450			MOV	LSDLY,R5	:	*,SSTMP1
16011	056446	001404				BEQ	7\$:	*,SSTMP1
16012	056450	005066	000014		6\$:	CLR	14(SP)	:	SSTMP
16013	056454	005305				DEC	R5	:	SSTMP1
16014	056456	001374				BNE	6\$:	SSTMP1
16015	056460	005304			7\$:	DEC	R4	:	SSTMP2
16016	056462	000766				BR	5\$:	SSTMP2
16017	056464	012777	000020	137032	8\$:	BISB	#20,@ML.REG+120	:	
16018	056472	005203				INC	R3	:	CNT
16019	056474	020327	000177			CMP	R3,#177	:	CNT,*
16020	056500	003755				BLE	4\$:	CNT,*
16021	056502	152777	000040	136734		BISB	#40,@ML.REG+40	:	
16022	056510	016705	137316			MOV	ML.DUT,R5	:	
16023	056514	042705	177770			BIC	#177770,R5	:	
16024	056520	142777	000007	136716		BICB	#7,@ML.REG+40	:	
16025	056526	150577	136712			BISB	R5,@ML.REG+40	:	
16026	056532	152777	000010	136764		BISB	#10,@ML.REG+120	:	
16027	056540	012777	177400	136646		MOV	#-400,@ML.REG+10	:	7449
16028	056546	012777	014022	136650		MOV	#10.BUF,@ML.REG+20	:	7450
16029	056554	010177	136654			MOV	R1,@ML.REG+30	:	7451
16030	056560	012777	000071	136616		MOV	#71,@ML.REG	:	7452
16031	056566	012704	000001			MOV	#1,R4	:	7453
16032	056572	001411			9\$:	BEQ	12\$:	*,SSTMP2
16033	056574	016705	123316			MOV	LSDLY,R5	:	*,SSTMP1
16034	056600	001404				BEQ	11\$:	*,SSTMP1
16035	056602	005066	000014		10\$:	CLR	14(SP)	:	SSTMP
16036	056606	005305				DEC	R5	:	SSTMP1
16037	056610	001374				BNE	10\$:	SSTMP1
16038	056612	005304			11\$:	DEC	R4	:	SSTMP2
16039	056614	000766				BR	9\$:	SSTMP2
16040	056616	005003			12\$:	CLR	R3	:	WD.CNT
16041	056620	017767	137010	136514	13\$:	MOV	@ML.REG+230,PD.TEMP	:	7456
16042	056626	152777	000020	136670		BISB	#20,@ML.REG+120	:	7458
16043	056634	012704	000001			MOV	#1,R4	:	7459
16044	056640	001411			14\$:	BEQ	17\$:	*,SSTMP2
16045	056642	016705	123250			MOV	LSDLY,R5	:	*,SSTMP1
16046	056646	001404				BEQ	16\$:	*,SSTMP1
16047	056650	005066	000014		15\$:	CLR	14(SP)	:	SSTMP
16048	056654	005305				DEC	R5	:	SSTMP1
16049	056656	001374				BNE	15\$:	SSTMP1
16050	056660	005304			16\$:	DEC	R4	:	SSTMP2
16051	056662	000766				BR	14\$:	SSTMP2
16052	056664	017767	136704	134502	17\$:	MOV	@ML.REG+170,D1.TEMP	:	


```

16110      ;ML4AD
16111      ;
16112      ; TEST CODE SECTION
16113 057130 012746 000002      MOV #2,-(SP)
16114 057134 010600      MOV SP,R0
16115 057136 104414      TRAP 14 ; SP,*
16116 057140 012766 000001 000026  MOV #1,26(SP)
16117 057146 062706 000016      ADD #16,SP ; *,DODU.FLG
16118 057152 104467      22$: TRAP 67 ;
16119 057154 006000      ROR R0 ;
16120 057156 103002      BHIS 23$
16121 057160 000167 177142      JMP 2$
16122 057164 026627 000010 000001 23$: CMP 10(SP),#1 ; DODU.FLG,*
16123 057172 001004      BNE 24$ ;
16124 057174 016700 136630      MOV ML,LUN,R0 ;
16125 057200 104451      TRAP 51 ;
16126 057202 104444      TRAP 44 ;
16127 057204 066601 000004 24$: ADD 4(SP),R1 ; *,ARR.SEL
16128 057210 020166 000006 25$: CMP R1,6(SP) ; ARR.SEL,*
16129 057214 101002      BHI 26$
16130 057216 000167 177100      JMP 1$
16131 057222 062706 000016 26$: ADD #16,SP ;
16132 057226 000207      RTS PC ;
16133
16134      ; Routine Size: 244 words
16135      ; Maximum stack depth per invocation: 20 words
16140
16141
16145
16146      .SBTTL T28 TEST CODE SECTION
16150 057230      T28::
16151 057230 004767 177024 1$: JSR PC,ST28 ;
16152 057234 104466      TRAP 66 ;
16153 057236 006000      ROR R0 ;
16154 057240 103773      BLO 1$ ;
16155 057242 000207      RTS PC ;
16156
16157      ; Routine Size: 6 words
16158      ; Maximum stack depth per invocation: 0 words
16163
16164 : 7501 !<BLF/PAGE>

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (81)

16166 :ML4AD
16167 :
16168 :
16169 : 7502
16170 : 7503
16171 : 7504
16172 : 7505
16173 : 7506
16174 : 7507
16175 : 7508
16176 : 7509
16177 : 7510
16178 : 7511
16179 : 7512
16180 : 7513
16181 : 7514
16182 : 7515
16183 : 7516
16184 : 7517
16185 : 7518
16186 : 7519
16187 : 7520
16188 : 7521
16189 : 7522
16190 : 7523
16191 : 7524
16192 : 7525
16193 : 7526
16194 : 7527
16195 : 7528
16196 : 7529
16197 : 7530
16198 : 7531
16199 : 7532
16200 : 7533
16201 : 7534
16202 : 7535
16203 : 7536
16204 : 7537
16205 : 7538
16206 : 7539
16207 : 7540
16208 : 7541
16209 : 7542
16210 : 7543
16211 : 7544
16212 : 7545
16213 : 7546
16214 : 7547
16215 : 7548
16216 : 7549
16217 : 7550
16218 : 7551
16219 : 7552
16220 : 7553

TEST CODE SECTION

! 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

16222	:	ML4AD			
16223	:		TEST CODE SECTION		
16224	:				
16225	:	7554	PASS CNT,		!PASS COUNT
16226	:	7555	NIB PAT,		!NIBBLE PATTERN
16227	:	7556	DONE FLG,		!DONE FLAG
16228	:	7557	ERR FLG,		!ERROR FLAG
16229	:	7558	SECTOR_NO;		!SECTOR NUMBER
16230	:	7559			
16231	:	7560	BGNSUB;		
16232	:	7561	PASS CNT = -1;		
16233	:	7562	SECTOR_NO = ZEROES;		
16234	:	7563	DONE_FLG = ZERO;		
16235	:	7564			
16236	:	7565	do		!THIS LOOP DETERMINES WHEN TO STOP
16237	:	7566	begin		
16238	:	7567			
16239	:	7568	do		!THIS LOOP RUNS THE PATTERNS
16240	:	7569	begin		
16241	:	7570	PASS CNT = .PASS_CNT + 1;		!INCREMENT THE PASS COUNT
16242	:	7571	CLR_MBUS;		
16243	:	7572	DAT_DM = ONE;		!SET DATA DIAG MODE
16244	:	7573	MLDA = .SECTOR_NO;		!LOAD SECTOR NUMBER IN DSA
16245	:	7574	MLWC = not 255;		!LOAD WORD COUNT
16246	:	7575	MLBA = IO_BUF;		!LOAD UBUS ADRS
16247	:	7576			
16248	:	7577	case .PASS_CNT from 0 to 3 of		!SELECT A NIBBLE PATTERN
16249	:	7578	set		
16250	:	7579			
16251	:	7580	[0] :		
16252	:	7581	NIB_PAT = %0'000000';		!ZEROES
16253	:	7582			
16254	:	7583	[1] :		
16255	:	7584	NIB_PAT = %0'17';		!ONES
16256	:	7585			
16257	:	7586	[2] :		
16258	:	7587	NIB_PAT = %0'12';		!ALTERNATING ONE'S, ZEROES
16259	:	7588			
16260	:	7589	[3] :		
16261	:	7590	NIB_PAT = %0'15'		!COMPLIMENT ONE'S, ZEROES
16262	:	7591	tes;		
16263	:	7592			
16264	:	7593	incr LD_CNT from 0 to 9 do		!LOAD NIBBLE PATTERN INTO NIBBLE SAVE
16265	:	7594	LD_LNG_WRD (.LD_CNT, .NIB_PAT);		
16266	:	7595			
16267	:	7596	WRT_LNG_WRD;		!LOAD THE DATA DIAG REGISTERS WITH NIBBLE SAVE
16268	:	7597	MLCS1 = write;		!DO A WRITE FUNCTION
16269	:	7598			
16270	:	7599	incr WRT_CNT from 0 to 127 do		!WRITE PATTERN INTO THIS BLOCK
16271	:	7600	begin		
16272	:	7601	DELAY (ONE_US);		
16273	:	7602	DAT_CLK = ONE;		
16274	:	7603	end;		
16275	:	7604			
16276	:	7605	CLR_MBUS;		

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (81)

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (81)

16278 :ML4AD
16279 :
16280 :
16281 :
16282 :
16283 :
16284 :
16285 :
16286 :
16287 :
16288 :
16289 :
16290 :
16291 :
16292 :
16293 :
16294 :
16295 :
16296 :
16297 :
16298 :
16299 :
16300 :
16301 :
16302 :
16303 :
16304 :
16305 :
16306 :
16307 :
16308 :
16309 :
16310 :
16311 :
16312 :
16313 :
16314 :
16315 :
16316 :
16317 :
16318 :
16319 :
16320 :
16321 :
16322 :
16323 :
16324 :
16325 :
16326 :
16327 :
16328 :
16329 :
16330 :
16331 :
16332 :

```

TEST CODE SECTION

    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_PTR, 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
    !GOOD BLOCK GETS THIS SECTOR NO
else
    begin
        SECTOR_NO = .SECTOR_NO + .RAS_INC;      !NO
        PASS_CNT = -1;      !INCREMENT ROW NO
    end
    !RESET PASS COUNT

end

!VER CZMLAD CHANGED TEST TO UNSIGNED TEST
until (.DONE_FLG IS_SET ) or (.SECTOR_NO eqlU .LST_ARR + .ARR_INC);
    
```


					TEST CODE SECTION			29-Mar-1982 16:23:04	TOPS
								29-Mar-1982 16:21:03	PA:<
16390									
16391					:ML4AD				
16392					:				
16393	057414	000402				BR	8\$		
16394	057416	012701	000015		7\$:	MOV	#15,R1	*.NIB.PAT	7577
16395	057422	005005			8\$:	CLR	R5	LD.CNT	7590
16396	057424	010546			9\$:	MOV	R5,-(SP)	LD.CNT,*	7593
16397	057426	010146				MOV	R1,-(SP)	NIB.PAT,*	7594
16398	057430	004767	141720			JSR	PC,LD.LNG.WRD		
16399	057434	022626				CMP	(SP)+,(SP)+		
16400	057436	005205				INC	R5	LD.CNT	7593
16401	057440	020527	000011			CMP	R5,#11	LD.CNT,*	
16402	057444	003767				BLE	9\$		
16403	057446	016777	133722	136120		MOV	D1.TEMP,@ML.REG+170		
16404	057454	016777	133716	136122		MOV	D2.TEMP,@ML.REG+200		7594
16405	057462	016777	133712	136074		MOV	E2.TEMP,@ML.REG+160		
16406	057470	012777	000061	135706		MOV	#61,@ML.REG		
16407	057476	005003				CLR	R3	WRT.CNT	7597
16408	057500	012704	000001		10\$:	MOV	#1,R4	*.SSTMP2	7599
16409	057504	001411			11\$:	BEQ	14\$		7601
16410	057506	016705	122404			MOV	LSDLY,R5	*.SSTMP1	
16411	057512	001404				BEQ	13\$		
16412	057514	005066	000012		12\$:	CLR	12(SP)	SSTMP	
16413	057520	005305				DEC	R5	SSTMP1	
16414	057522	001374				BNE	12\$		
16415	057524	005304			13\$:	DEC	R4	SSTMP2	
16416	057526	000766				BR	11\$		
16417	057530	152777	000020	135766	14\$:	BISB	#20,@ML.REG+120		
16418	057536	005203				INC	R3	WRT.CNT	7602
16419	057540	020327	000177			CMP	R3,#177	WRT.CNT,*	7599
16420	057544	003755				BLE	10\$		
16421	057546	152777	000040	135670		BISB	#40,@ML.REG+40		
16422	057554	016705	136252			MOV	ML.DUT,R5		7603
16423	057560	042705	177770			BIC	#177770,R5		
16424	057564	142777	000007	135652		BICB	#7,@ML.REG+40		
16425	057572	150577	135646			BISB	R5,@ML.REG+40		
16426	057576	152777	000010	135720		BISB	#10,@ML.REG+120		
16427	057604	010277	135524			MOV	R2,@ML.REG+30	SECTOR.NO,*	7606
16428	057610	012777	177400	135576		MOV	#-400,@ML.REG+10		7607
16429	057616	012777	014022	135600		MOV	#10.BUF,@ML.REG+20		7608
16430	057624	012777	000071	135552		MOV	#71,@ML.REG		7609
16431	057632	012704	000001			MOV	#1,R4	*.SSTMP2	7610
16432	057636	001411			15\$:	BEQ	18\$		7611
16433	057640	016705	122252			MOV	LSDLY,R5	*.SSTMP1	
16434	057644	001404				BEQ	17\$		
16435	057646	005066	000012		16\$:	CLR	12(SP)	SSTMP	
16436	057652	005305				DEC	R5	SSTMP1	
16437	057654	001374				BNE	16\$		
16438	057656	005304			17\$:	DEC	R4	SSTMP2	
16439	057660	000766				BR	15\$		
16440	057662	012766	177777	000006	18\$:	MOV	#-1,6(SP)	*.WRD.CNT	7612
16441	057670	005266	000006		19\$:	INC	6(SP)	WRD.CNT	7616
16442	057674	017767	135734	135440		MOV	@ML.REG+230,PD.TEMP		7617
16443	057702	152777	000020	135614		BISB	#20,@ML.REG+120		7618
16444	057710	012704	000001			MOV	#1,R4	*.SSTMP2	7619

16446									
16447									
16448									
16449	057714	001411		20\$:	BEQ	23\$			
16450	057716	016705	122174		MOV	L\$DLY,R5	: *,SSTMP1		
16451	057722	001404			BEQ	22\$			
16452	057724	005066	000012	21\$:	CLR	12(SP)	: SSTMP		
16453	057730	005305			DEC	R5	: SSTMP1		
16454	057732	001374			BNE	21\$			
16455	057734	005304		22\$:	DEC	R4	: SSTMP2		
16456	057736	000766			BR	20\$			
16457	057740	017767	135630	23\$:	MOV	@ML.REG+170,D1.TEMP			
16458	057746	017767	135632		MOV	@ML.REG+200,D2.TEMP			
16459	057754	017767	135604		MOV	@ML.REG+160,E2.TEMP			
16460	057762	012716	177777		MOV	#-1,(SP)			
16461	057766	005216		24\$:	INC	(SP)	: *,NIB.PTR	7621	
16462	057770	011605			MOV	(SP),R5	: NIB.PTR	7625	
16463	057772	006205			ASR	R5	: NIB.PTR,*	7627	
16464	057774	006205			ASR	R5			
16465	057776	006205			ASR	R5			
16466	060000	062705	015342		ADD	#PD.TEMP,R5			
16467	060004	010546			MOV	R5,-(SP)			
16468	060006	016646	000002		MOV	2(SP),-(SP)			
16469	060012	042716	177770		BIC	#177770,(SP)	: NIB.PTR,*		
16470	060016	012746	000001		MOV	#1,-(SP)			
16471	060022	005046			CLR	-(SP)			
16472	060024	004767	123604		JSR	PC,BL\$GT2			
16473	060030	062706	000010		ADD	#10,SP			
16474	060034	005700			TST	R0			
16475	060036	001011			BNE	25\$			
16476	060040	011646			MOV	(SP),-(SP)			
16477	060042	010146			MOV	R1,-(SP)	: NIB.PTR,*	7629	
16478	060044	012746	000016		MOV	#16,-(SP)	: NIB.PAT,*		
16479	060050	060616			ADD	SP,(SP)			
16480	060052	004767	140320		JSR	PC,TST.LNG.WRD	: ERR.FLG,*		
16481	060056	062706	000006		ADD	#6,SP			
16482	060062	016605	000010	25\$:	MOV	10(SP),R5	: ERR.FLG,*		
16483	060066	032705	000001		BIT	#1,R5		7632	
16484	060072	001012			BNE	26\$			
16485	060074	021627	000011		CMP	(SP),#11	: NIB.PTR,*		
16486	060100	001332			BNE	24\$			
16487	060102	032705	000001		BIT	#1,R5			
16488	060106	001004			BNE	26\$		7635	
16489	060110	026627	000006	000177	CMP	6(SP),#177	: WRD.CNT,*		
16490	060116	001264			BNE	19\$			
16491	060120	005004			CLR	R4			
16492	060122	026627	000002	000003	26\$:	CMP	2(SP),#3	: PASS.CNT,*	7638
16493	060130	001002			BNE	27\$			
16494	060132	005204			INC	R4			
16495	060134	000403			BR	28\$			
16496	060136	020527	000001	27\$:	CMP	R5,#1			
16497	060142	001030			BNE	31\$			
16498	060144	006004		28\$:	ROR	R4			
16499	060146	103010			BCC	29\$		7640	
16500	060150	005705			TST	R5			

16502									
16503									
16504									
16505	060152	001006							
16506	060154	012766	000001	000004					
16507	060162	010267	133620						
16508	060166	000405							
16509	060170	066702	135152		29\$:				
16510	060174	012766	177777	000002					
16511	060202	026627	000004	000001	30\$:				
16512	060210	001410							
16513	060212	016705	133602						
16514	060216	066705	133562						
16515	060222	020205							
16516	060224	001402			31\$:				
16517	060226	000167	177040						
16518	060232	104467			32\$:				
16519	060234	006000							
16520	060236	103002							
16521	060240	000167	177010						
16522	060244	016705	133550		33\$:				
16523	060250	066705	133530						
16524	060254	020205							
16525	060256	001026							
16526	060260	104455							
16527	060262	000117							
16528	060264	013114							
16529	060266	026302							
16530	060270	012746	012360						
16531	060274	012746	012006						
16532	060300	012746	010664						
16533	060304	012746	010256						
16534	060310	012746	000004						
16535	060314	010600							
16536	060316	104414							
16537	060320	016700	135504						
16538	060324	104451							
16539	060326	104444							
16540	060330	062706	000012						
16541	060334	062706	000014		34\$:				
16542	060340	000207							
16543									
16544									
16545									
16550									
16551									
16555									
16556									

: Routine Size: 287 words
 : Maximum stack depth per invocation: 17 words

.SBTTL T29 TEST CODE SECTION

16558
16559
16560
16564 060342
16565 060342 004767 176676
16566 060346 104466
16567 060350 006000
16568 060352 103773
16569 060354 000207
16570
16571
16572
16577
16578
16579 ; 7674 !<BLF/PAGE>

:ML4AD
:
TEST CODE SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

T29::
18: JSR PC,ST29
TRAP 66
ROR R0
BLO 18
RTS PC

7671

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (82)

16581 :ML4AD
16582 :
16583 :
16584 :
16585 :
16586 :
16587 :
16588 :
16589 :
16590 :
16591 :
16592 :
16593 :
16594 :
16595 :
16596 :
16597 :
16598 :
16599 :
16600 :
16601 :
16602 :
16603 :
16604 :
16605 :
16606 :
16607 :
16608 :
16609 :
16610 :
16611 :
16612 :
16613 :
16614 :
16615 :
16616 :
16617 :
16618 :
16619 :
16620 :
16621 :
16622 :
16623 :
16624 :
16625 :
16626 :
16627 :
16628 :
16629 :
16630 :
16631 :
16632 :
16633 :
16634 :
16635 :

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 30

TEST NAME: REFRESH TIMING TEST

TEST DESCRIPTION:

TEST THE MEMORY ARRAY MODULES
REFRESH TIMING AND CONTROL LOGIC
REG BY:

1. WRITING ALL ONES INTO THE GOOD BLOCK.
2. DELAY FOR 2 MS
3. READ THE GOOD BLOCK FOR ONES. KEEP COUNT OF BAD NIBBLES ENCOUNTERED
4. ALLOW A BAD NIBBLE THRESHOLD OF 20 BAD NIBBLES OUT OF THE 1280 NIBBLES TESTED.
5. IF THRESHOLD IS EXCEEDED THEN REPORT ERROR AND DROP THE UNIT.

IMPLICIT INPUTS:

PD TEMP:
A BIT VECTOR OF 16 BITS WHERE
THE READ PROM DATA IS STORED
AND ACCESSED FROM.

Local

TST_PAT,
ERR_CNT,
ERR_FLG:

!TEST PATTERN
!ERROR COUNT
!ERROR FLAG

BGNSUB:

CLR_MBUS:

ERR_CNT = ZERO:

TST_PAT = ONES:

MLD1 = .TST_PAT:

MLD2 = .TST_PAT:

MLE2 = .TST_PAT:

!LOAD DATA DIAG REGS WITH TST PAT

```

16637 :ML4AD
16638 :
16639 :
16640 : 7727 DAT_DM_XFER ();
16641 : 7728 MLCS1 = write;
16642 : 7729
16643 : 7730 incr WRD_CNT from 0 to 127 do
16644 : 7731   begin
16645 : 7732   DELAY (ONE_US);
16646 : 7733   DAT_CLK = ONE;
16647 : 7734   end;
16648 : 7735
16649 : 7736 CLR_MBUS;
16650 : 7737 DAT_DM_XFER ();
16651 : 7738 MLCS1 = read;
16652 : 7739 DELAY (TWO_TH_US);
16653 : 7740
16654 : 7741 incr WRD_CNT from 0 to 127 do
16655 : 7742   begin
16656 : 7743   PD_TEMP = .MLPD;
16657 : 7744   DAT_CLK = ONE;
16658 : 7745   DELAY (ONE_US);
16659 : 7746   RD_LNG_WRD;
16660 : 7747
16661 : 7748   incr NIB_PTR from 0 to 9 do
16662 : 7749     begin
16663 : 7750
16664 : 7751     if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
16665 : 7752
16666 : 7753     !FIND GOOD NIBBLES AND COMPARE THEM
16667 : 7754
16668 : 7755     if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1; !INCREMENT ERROR COUNT IF ERR_FLG IS SET
16669 : 7756
16670 : 7757     end;
16671 : 7758
16672 : 7759   end;
16673 : 7760
16674 : 7761 ENDSUB;
16675 : 7762
16676 : 7763 if .ERR_CNT gtr 20
16677 : 7764 then
16678 : 7765   begin
16679 : 7766   ERRDF (80, ASYNC, DUMPER);
16680 : 7767   PRINTB (FIV_FMT, WRD_22, PHR_4, WRD_12, FNC_16, WRD_48);
16681 : 7768   DODU (.ML_LDN);
16682 : 7769   DOCLN;
16683 : 7770   end;
16684 : 7771
16685 : 7772 ENDTST;
16689 :
16690 :

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (82)

!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
!ALLOW 20 NIBBLES TO FAIL
!ERROR IF GTR 20
.SBTTL \$T30 TEST CODE SECTION

Address	OpCode	Op1	Op2	Op3	Op4	Label	Instruction	Comments	Seq
16698	060356	004167	124230			ST30:	JSR R1,\$SAVE5	:	
16699	060362	024646					CMP -(SP),-(SP)	:	7673
16700	060364	104402				1\$:	TRAP 2	:	
16701	060366	152777	000040	135050			BISB #40,@ML.REG+40	:	7718
16702	060374	016703	135432				MOV ML,DUT,R3	:	7720
16703	060400	042703	177770				BIC #177770,R3	:	
16704	060404	142777	000007	135032			BICB #7,@ML.REG+40	:	
16705	060412	150377	135026				BISB R3,@ML.REG+40	:	
16706	060416	005005					CLR R5	:	
16707	060420	012704	177777				MOV #-1,R4	:	ERR.CNT 7722
16708	060424	010477	135144				MOV R4,@ML.REG+170	:	*.TST.PAT 7723
16709	060430	010477	135150				MOV R4,@ML.REG+200	:	TST.PAT,* 7724
16710	060434	010477	135124				MOV R4,@ML.REG+160	:	TST.PAT,* 7725
16711	060440	004767	135746				JSR PC,DAT.DM.XFER	:	TST.PAT,* 7726
16712	060444	012777	000061	134732			MOV #61,@ML.REG	:	7727
16713	060452	005001					CLR R1	:	7728
16714	060454	012702	000001			2\$:	MOV #1,R2	:	WRD.CNT 7730
16715	060460	001411				3\$:	BEQ 6\$:	*.SSTMP2 7732
16716	060462	016703	121430				MOV LSDLY,R3	:	*.SSTMP1
16717	060466	001404					BEQ 5\$:	
16718	060470	005066	000002			4\$:	CLR 2(SP)	:	SSTMP
16719	060474	005303					DEC R3	:	SSTMP1
16720	060476	001374					BNE 4\$:	
16721	060500	005302				5\$:	DEC R2	:	SSTMP2
16722	060502	000766					BR 3\$:	
16723	060504	152777	000020	135012		6\$:	BISB #20,@ML.REG+120	:	
16724	060512	005201					INC R1	:	WRD.CNT 7733
16725	060514	020127	000177				CMP R1,#177	:	WRD.CNT,* 7730
16726	060520	003755					BLE 2\$:	
16727	060522	152777	000040	134714			BISB #40,@ML.REG+40	:	
16728	060530	016703	135276				MOV ML,DUT,R3	:	7734
16729	060534	042703	177770				BIC #177770,R3	:	
16730	060540	142777	000007	134676			BICB #7,@ML.REG+40	:	
16731	060546	150377	134672				BISB R3,@ML.REG+40	:	
16732	060552	004767	135634				JSR PC,DAT.DM.XFER	:	
16733	060556	012777	000071	134620			MOV #71,@ML.REG	:	7737
16734	060564	012702	003720				MOV #3720,R2	:	7738
16735	060570	001411				7\$:	BEQ 10\$:	*.SSTMP2 7739
16736	060572	016703	121320				MOV LSDLY,R3	:	*.SSTMP1
16737	060576	001404					BEQ 9\$:	
16738	060600	005066	000002			8\$:	CLR 2(SP)	:	SSTMP
16739	060604	005303					DEC R3	:	SSTMP1
16740	060606	001374					BNE 8\$:	
16741	060610	005302				9\$:	DEC R2	:	SSTMP2
16742	060612	000766					BR 7\$:	
16743	060614	005001				10\$:	CLR R1	:	WRD.CNT 7741
16744	060616	017767	135012	134516		11\$:	MOV @ML.REG+230,PD.TEMP	:	7743
16745	060624	152777	000020	134672			BISB #20,@ML.REG+120	:	7744
16746	060632	012702	000001				MOV #1,R2	:	7745
16747	060636	001411				12\$:	BEQ 15\$:	*.SSTMP2

Address	Hex	Dec	Label	Op	Operand	Comment	Page
16749			:ML4AD				
16750			:				
16751							
16752	060640	016703	121252	MOV	LSDLY,R3	: *,SSTMP1	
16753	060644	001404		BEQ	14\$		
16754	060646	005066	000002	13\$: CLR	2(SP)	: SSTMP	
16755	060652	005303		DEC	R3	: SSTMP1	
16756	060654	001374		BNE	13\$		
16757	060656	005302		14\$: DEC	R2	: SSTMP2	
16758	060660	000766		BR	12\$		
16759	060662	017767	134706 132504	15\$: MOV	2ML.REG+170,D1.TEMP		
16760	060670	017767	134710 132500	MOV	2ML.REG+200,D2.TEMP		
16761	060676	017767	134662 132474	MOV	2ML.REG+160,E2.TEMP		
16762	060704	005002		CLR	R2	: NIB.PTR	
16763	060706	010203		16\$: MOV	R2,R3	: NIB.PTR,*	7748
16764	060710	006203		ASR	R3		7751
16765	060712	006203		ASR	R3		
16766	060714	006203		ASR	R3		
16767	060716	062703	015342	ADD	#PD.TEMP,R3		
16768	060722	010346		MOV	R3,-(SP)		
16769	060724	010246		MOV	R2,-(SP)	: NIB.PTR,*	
16770	060726	042716	177770	BIC	#177770,(SP)		
16771	060732	012746	000001	MOV	#1,-(SP)		
16772	060736	005046		CLR	-(SP)		
16773	060740	004767	122670	JSR	PC,BLSGT2		
16774	060744	062706	000010	ADD	#10,SP		
16775	060750	005700		TST	R0		
16776	060752	001011		BNE	17\$		
16777	060754	010246		MOV	R2,-(SP)	: NIB.PTR,*	
16778	060756	010446		MOV	R4,-(SP)	: TST.PAT,*	
16779	060760	012746	000006	MOV	#6,-(SP)		
16780	060764	060616		ADD	SP,(SP)	: ERR.FLG,*	
16781	060766	004767	137404	JSR	PC,TST.LNG.WRD		
16782	060772	062706	000006	ADD	#6,SP		
16783	060776	021627	000001	17\$: CMP	(SP),#1	: ERR.FLG,*	7755
16784	061002	001001		BNE	18\$		
16785	061004	005205		INC	R5	: ERR.CNT	
16786	061006	005202		18\$: INC	R2	: NIB.PTR	7748
16787	061010	020227	000011	CMP	R2,#11	: NIB.PTR,*	
16788	061014	003734		BLE	16\$		
16789	061016	005201		INC	R1	: WRD.CNT	7741
16790	061020	020127	000177	CMP	R1,#177	: WRD.CNT,*	
16791	061024	003674		BLE	11\$		
16792	061026	104467		TRAP	67	:	7759
16793	061030	006000		ROR	R0		
16794	061032	103002		BHIS	19\$		
16795	061034	000167	177324	JMP	1\$		
16796	061040	020527	000024	19\$: CMP	R5,#24	: ERR.CNT,*	7763
16797	061044	003432		BLE	20\$		
16798	061046	104455		TRAP	55	:	7766
16799	061050	000120		.WORD	120		
16800	061052	012706		.WORD	ASync		
16801	061054	026302		.WORD	DUMPER		
16802	061056	012746	011250	MOV	#WRD.48,-(SP)		
16803	061062	012746	012372	MOV	#FNC.16,-(SP)		7767

```

16805
16806
16807
16808 061066 012746 010650
16809 061072 012746 011676
16810 061076 012746 010752
16811 061102 012746 010304
16812 061106 012746 000006
16813 061112 010600
16814 061114 104414
16815 061116 016700 134706
16816 061122 104451
16817 061124 104444
16818 061126 062706 000016
16819 061132 022626
16820 061134 000207
16821
16822
16823
16828
16829
16833
16834
16838 061136
16839 061136 004767 177214
16840 061142 104466
16841 061144 006000
16842 061146 103773
16843 061150 000207
16844
16845
16846
16851
16852
16853 ; 7773 !<BLF/PAGE>

```

```

:ML4AD
:
TEST CODE SECTION
MOV #WRD.12,-(SP)
MOV #PHR.4,-(SP)
MOV #WRD.22,-(SP)
MOV #FIV.FMT,-(SP)
MOV #6,-(SP)
MOV SP,R0
TRAP 14 ; SP,*
MOV ML.LUN,R0
TRAP 51 ;
TRAP 44 ;
ADD #16,SP
20$: CMP (SP)+,(SP)+ ;
RTS PC ;

```

```

; Routine Size: 184 words
; Maximum stack depth per invocation: 15 words

```

```

.SBTTL T30 TEST CODE SECTION
T30::
1$: JSR PC,$T30 ;
TRAP 66 ;
ROR R0
BLO 1$
RTS PC

```

```

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

```

```

7768
7765
7673

```

7770

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (83)

16855 : ML4AD
 16856 :
 16857 :
 16858 :
 16859 :
 16860 :
 16861 :
 16862 :
 16863 :
 16864 :
 16865 :
 16866 :
 16867 :
 16868 :
 16869 :
 16870 :
 16871 :
 16872 :
 16873 :
 16874 :
 16875 :
 16876 :
 16877 :
 16878 :
 16879 :
 16880 :
 16881 :
 16882 :
 16883 :
 16884 :
 16885 :
 16886 :
 16887 :
 16888 :
 16889 :
 16890 :
 16891 :
 16892 :
 16893 :
 16894 :
 16895 :
 16896 :
 16897 :
 16898 :
 16899 :
 16900 :
 16901 :
 16902 :
 16903 :
 16904 :
 16905 :
 16906 :
 16907 :
 16908 :
 16909 :

TEST CODE SECTION

7774 !
 7775 BGNTST;

7776 !++

7777
 7778 TEST NUMBER: TST 31

7779
 7780 TEST NAME: ADDRESS COUNTER TEST

7781 TEST DESCRIPTION:

7782 TEST THE ABILITY OF THE ADDRESS
 7783 COUNTER TO SUCCESSFULLY COUNT
 7784 FROM BLOCK ZERO THROUGH THE
 7785 DEVICES LAST BLOCK BY:

- 7786 1. WRITING THE LAST BLOCK WITH ONES PATTERN.
- 7787 2. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK UP TO THE
 7788 LAST BLOCK ADRS MINUS ONE.
 7789 READ THE LAST BLOCK FOR ONES AND ERROR IF ZEROES.
- 7790 3. STARTING AT BLOCK ZERO WRITE ZEROES INTO ALL BLOCK THROUGH THE LAST
 7791 BLOCK.
 7792 READ THE LAST BLOCK FOR ZEROES AND ERROR IF STILL ONES.

7793 IMPLICIT INPUTS:

7794 PD TEMP:
 7795 A BITVECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND ACCESSED FROM.

7796 Local

7797 ERR_CNT, !ERROR THRESHOLD COUNT
 7798 ERR_FLG, !ERROR FLG
 7799 END_CNT, !ENDING SECTOR NUMBER
 7800 BG_PAT; !BACKGROUND PATTERN

7801 BGNSUB;

7802 CLR MBUS;

7803 BG_PAT = ONES;

7804 MLD1 = .BG_PAT;

7805 MLD2 = .BG_PAT;

7806 MLE2 = .BG_PAT;

7807 DAT_DM = ONE;

7808 LAST_BLK_XFER ();

7809 MLCST = write;

7810 incr WRD_CNT from 0 to 127 do

7811 begin

7812 DELAY (ONE US);

7813 DAT_CLK = ONE;

7814 end;

!BACKGROUND PAT OF ONES
 !LOAD DATA DIAGS WITH BG PAT

!SET DATA DIAG MODE
 !SET UP A LAST BLOCK XFERR
 !DO A WRITE FUNCTION

!WRITE THE LAST BLOCK WITH THE BG PATTERN

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 v2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (83)

```

16911 :ML4AD
16912 :
16913 : TEST CODE SECTION
16914 : 7826
16915 : 7827 CLR_MBUS;
16916 : 7828 DAT_DM = ONE;
16917 : 7829 LAST_BLK_XFER ();
16918 : 7830 MLCST = read;
16919 : 7831 DELAY (ONE_US);
16920 : 7832 ERR_CNT = ZERO;
16921 : 7833
16922 : 7834 incr WD_CNT from 0 to 127 do
16923 : 7835 begin
16924 : 7836 PD_TEMP = .MLPD;
16925 : 7837 DAT_CLK = ONE;
16926 : 7838 DELAY (ONE_US);
16927 : 7839 RD_LNG_WRD;
16928 : 7840
16929 : 7841 incr NIB_PTR from 0 to 9 do
16930 : 7842 begin
16931 : 7843
16932 : 7844 if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .BG_PAT, Err_FLG);
16933 : 7845
16934 : 7846
16935 : 7847
16936 : 7848 if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + 1; !INC ERROR COUNT IF ERROR DETECTED
16937 : 7849
16938 : 7850 end;
16939 : 7851
16940 : 7852 end;
16941 : 7853
16942 : 7854 if .ERR_CNT gtr 10
16943 : 7855 then
16944 : 7856 begin
16945 : 7857 ERRDF (81, INTER, DUMPER);
16946 : 7858 PRINTB (THR_FMT, PHR_4, WRD_13, FNC_22);
16947 : 7859 PRINTB (TWO_FMT, FNC_13, WRD_56);
16948 : 7860 EXIT_TST;
16949 : 7861 end;
16950 : 7862
16951 : 7863 END_CNT = .LST_BLK - 1;
16952 : 7864 MLD1 = ZEROES;
16953 : 7865 MLD2 = ZEROES;
16954 : 7866 MLE2 = ZEROES;
16955 : 7867
16956 : 7868 incr TWICE from 0 to 1 do
16957 : 7869 begin
16958 : 7870 CLR_MBUS;
16959 : 7871 ERR_CNT = ZERO;
16960 : 7872 DAT_DM = ONE;
16961 : 7873 FIRST_BLK_XFER ();
16962 : 7874 MLCST = write;
16963 : 7875
16964 : 7876 !
16965 : 7877 !VER CZMLAD CHANGED TEST TO UNSIGNED TEST

```

```

!SET DATA DIAG MODE
!SET UP A LAST BLOCK XFERR
!DO A READ FUNCTION

!CLEAR ERROR COUNT THRESHOLD

!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

!INC ERROR COUNT IF ERROR DETECTED

!IS ERROR THRESHOLD EXCEEDED

!ERROR IF EXCEEDED

!END AT LAST BLOCK -1
!LOAD DATA DIAG REG WITH COMP BG PAT

!REPEAT LOOP TWICE

!CLEAR THE ERROR COUNT THRESHOLD
!SET DATA DIAG MODE
!SET UP A FIRST BLOCK XFERR
!DO A WRITE FUNCTION

```


29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (83)

```

16967 :ML4AD
16968 :
16969 :
16970 : 7878 !
16971 : 7879 !
16972 : 7880 !
16973 : 7881 !
16974 : 7882 !
16975 : 7883 !
16976 : 7884 !
16977 : 7885 !
16978 : 7886 !
16979 : 7887 !
16980 : 7888 !
16981 : 7889 !
16982 : 7890 !
16983 : 7891 !
16984 : 7892 !
16985 : 7893 !
16986 : 7894 !
16987 : 7895 !
16988 : 7896 !
16989 : 7897 !
16990 : 7898 !
16991 : 7899 !
16992 : 7900 !
16993 : 7901 !
16994 : 7902 !
16995 : 7903 !
16996 : 7904 !
16997 : 7905 !
16998 : 7906 !
16999 : 7907 !
17000 : 7908 !
17001 : 7909 !
17002 : 7910 !
17003 : 7911 !
17004 : 7912 !
17005 : 7913 !
17006 : 7914 !
17007 : 7915 !
17008 : 7916 !
17009 : 7917 !
17010 : 7918 !
17011 : 7919 !
17012 : 7920 !
17013 : 7921 !
17014 : 7922 !
17015 : 7923 !
17016 : 7924 !
17020 :
17021 :

TEST CODE SECTION

incrU BLK_CNT from 0 to .END_CNT do
    !CLOCK THE ADDRESS COUNTER UP TO END_CNT
    incr ADRS_CNT from 0 to 127 do
        begin
            DAT_CLK = ONE;
        end;

    CLR_MBUS;
    DAT_DM = ONE;
    LAST_BLK_XFER ();
    MLCST = read;
    DELAY (ONE_US);
    !SET DATA DIAG MODE
    !SET UP A LAST BLOCK XFERR
    !DO A READ FUNCTION

    incr WD_CNT from 0 to 127 do
        !READ THE LAST BLOCK FOR BG PATTERN
        begin
            PD_TEMP = .MLPD;
            DAT_CLK = ONE;
            DELAY (ONE_US);
            !GET THE PROM DATA
            !CLOCK OUT DATA WORD
            RD_LNG_WRD;
            !READ DATA WORD

            incr NIB_PTR from 0 to 9 do
                !LOOK AT 10 NIBBLES
                begin
                    if .PD_TEMP [.NIB_PTR] IS_NOT_SET then TST_LNG_WRD (.NIB_PTR, .BG_PAT, ERR_FLG);
                    !FIND GOOD NIBBLES AND COMPARE THEM

                    if .ERR_FLG IS_SET then ERR_CNT = .ERR_CNT + ONE;
                    !INC ERROR COUNT IF ERROR DETCTED

                end;

            end;

            if .ERR_CNT gtr 10
            then
                !IS ERROR COUNT THRESHOLD EXCEEDED
                begin
                    !ERROR IF EXCEEDED
                    ERRDF (82, ASYNC, DUMPER);
                    PRINTB (THR_FMT, WRD_50, WRD_51, WRD_10);
                end;

            END_CNT = .END_CNT + 1;
            !NOW END AT THE LAST BLOCK
            BG_PAT = not .BG_PAT;
            !COMPLIMENT THE BG PATTERN AND REPEAT
        end;

    ENDSUB;
    ENDTST;

.SBTTL $T31 TEST CODE SECTION

```


				:ML4AD		TEST CODE SECTION		29-Mar-1982 16:23:04		TOPS	
				:				29-Mar-1982 16:21:03		PA:<	
17079											
17080											
17081											
17082	061436	152777	000020	13406C		BISB	#20,@ML.REG+120	:			7837
17083	061444	012703	000001			MOV	#1,R3	:	*,SSTMP2		7838
17084	061450	001411			12\$:	BEQ	15\$:			
17085	061452	016704	120440			MOV	LSDLY,R4	:	*,SSTMP1		
17086	061456	001404				BEQ	14\$:			
17087	061460	005066	000006		13\$:	CLR	6(SP)	:	SSTMP		
17088	061464	005304				DEC	R4	:	SSTMP1		
17089	061466	001374				BNE	13\$:			
17090	061470	005303			14\$:	DEC	R3	:	SSTMP2		
17091	061472	000766				BR	12\$:			
17092	061474	017767	134074	131672	15\$:	MOV	@ML.REG+170,D1.TEMP	:			
17093	061502	017767	134076	131666		MOV	@ML.REG+200,D2.TEMP	:			
17094	061510	017767	134050	131662		MOV	@ML.REG+160,E2.TEMP	:			
17095	061516	005004				CLR	R4	:	NIB.PTR		7841
17096	061520	010403			16\$:	MOV	R4,R3	:	NIB.PTR,*		7844
17097	061522	006203				ASR	R3	:			
17098	061524	006203				ASR	R3	:			
17099	061526	006203				ASR	R3	:			
17100	061530	062703	015342			ADD	#PD.TEMP,R3	:			
17101	061534	010346				MOV	R3,-(SP)	:			
17102	061536	010446				MOV	R4,-(SP)	:	NIB.PTR,*		
17103	061540	042716	177770			BIC	#177770,(SP)	:			
17104	061544	012746	000001			MOV	#1,-(SP)	:			
17105	061550	005046				CLR	-(SP)	:			
17106	061552	004767	122056			JSR	PC,BLSGT2	:			
17107	061556	062706	000010			ADD	#10,SP	:			
17108	061562	005700				TST	R0	:			
17109	061564	001011				BNE	17\$:			
17110	061566	010446				MOV	R4,-(SP)	:	NIB.PTR,*		
17111	061570	010146				MOV	R1,-(SP)	:	BG.PAT,*		
17112	061572	012746	000012			MOV	#12,-(SP)	:			
17113	061576	060616				ADD	SP,(SP)	:	ERR.FLG,*		
17114	061600	004767	136572			JSR	PC,TST.LNG.WRD	:			
17115	061604	062706	000006			ADD	#6,SP	:			
17116	061610	026627	000004	000001	17\$:	CMP	4(SP),#1	:	ERR.FLG,*		7848
17117	061616	001001				BNE	18\$:			
17118	061620	005216				INC	(SP)	:	ERR.CNT		
17119	061622	005204			18\$:	INC	R4	:	NIB.PTR		7841
17120	061624	020427	000011			CMP	R4,#11	:	NIB.PTR,*		
17121	061630	003733				BLE	16\$:			
17122	061632	005202				INC	R2	:	WD.CNT		7834
17123	061634	020227	000177			CMP	R2,#177	:	WD.CNT,*		
17124	061640	003673				BLE	11\$:			
17125	061642	021627	000012			CMP	(SP),#12	:	ERR.CNT,*		7854
17126	061646	003437				BLE	19\$:			
17127	061650	104455				TRAP	55	:			7857
17128	061652	000121				.WORD	121	:			
17129	061654	013114				.WORD	INTER	:			
17130	061656	026302				.WORD	DUMPER	:			
17131	061660	012746	012452			MOV	#FNC.22,-(SP)	:			7858
17132	061664	012746	010660			MOV	#WRD.13,-(SP)	:			
17133	061670	012746	011676			MOV	#PHR.4,-(SP)	:			

Address	Op1	Op2	Op3	Label	Code	Instruction	Comments	Address
17135								
17136								
17137								
17138	061674	012746	010256			MOV #THR.FMT,-(SP)		
17139	061700	012746	000004			MOV #4,-(SP)		
17140	061704	010600				MOV SP,R0		
17141	061706	104414				TRAP 14	: SP,*	
17142	061710	012716	011344			MOV #WRD.56,(SP)		
17143	061714	012746	012334			MOV #FNC.13,-(SP)		7859
17144	061720	012746	010246			MOV #TWO.FMT,-(SP)		
17145	061724	012746	000003			MOV #3,-(SP)		
17146	061730	010600				MOV SP,R0		
17147	061732	104414				TRAP 14	: SP,*	
17148	061734	104463				TRAP 63		
17149	061736	062706	000020			ADD #20,SP		
17150	061742	000167	000560			JMP 38\$		7854
17151	061746	016766	132042	000002 19\$:		MOV LST.BLK,2(SP)		7856
17152	061754	005366	000002			DEC 2(SP)	: * ,END.CNT	7863
17153	061760	005077	133610			CLR @ML.REG+170	: END.CNT	
17154	061764	005077	133614			CLR @ML.REG+200		7864
17155	061770	005077	133570			CLR @ML.REG+160		7865
17156	061774	005005				CLR R5	: TWICE	7866
17157	061776	152777	000040	133440 20\$:		BISB #40,@ML.REG+40		7868
17158	062004	016704	134022			MOV ML.DUT,R4		7869
17159	062010	042704	177770			BIC #177770,R4		
17160	062014	142777	000007	133422		BICB #7,@ML.REG+40		
17161	062022	150477	133416			BISB R4,@ML.REG+40		
17162	062026	005016				CLR (SP)	: ERR.CNT	7871
17163	062030	152777	000010	133466		BISB #10,@ML.REG+120		7872
17164	062036	004767	136220			JSR PC,FIRST.BLK.XFER		7873
17165	062042	012777	000061	133334		MOV #61,@ML.REG		7874
17166	062050	005003				CLR R3	: BLK.CNT	7879
17167	062052	000411				BR 23\$		
17168	062054	005004				CLR R4	: ADRS.CNT	7881
17169	062056	152777	000020	133440 21\$:		BISB #20,@ML.REG+120		7883
17170	062064	005204		22\$:		INC R4	: ADRS.CNT	7881
17171	062066	020427	000177			CMP R4,#177	: ADRS.CNT,*	
17172	062072	003771				BLE 22\$		
17173	062074	005203				INC R3	: BLK.CNT	7879
17174	062076	020366	000002	23\$:		CMP R3,2(SP)	: BLK.CNT,END.CNT	
17175	062102	101764				BLOS 21\$		
17176	062104	152777	000040	133332		BISB #40,@ML.REG+40		7884
17177	062112	016704	133714			MOV ML.DUT,R4		
17178	062116	042704	177770			BIC #177770,R4		
17179	062122	142777	000007	133314		BICB #7,@ML.REG+40		
17180	062130	150477	133310			BISB R4,@ML.REG+40		
17181	062134	152777	000010	133362		BISB #10,@ML.REG+120		
17182	062142	004767	136176			JSR PC,LAST.BLK.XFER		7887
17183	062146	012777	000071	133230		MOV #71,@ML.REG		7888
17184	062154	012703	000001			MOV #1,R3	: *,SSTMP2	7889
17185	062160	001411		24\$:		BEQ 27\$		7890
17186	062162	016704	117730			MOV LSDLY,R4	: *,SSTMP1	
17187	062166	001404				BEQ 26\$		
17188	062170	005066	000006	25\$:		CLR 6(SP)	: SSTMP	
17189	062174	005304				DEC R4	: SSTMP1	

Address	OpCode	Operand 1	Operand 2	Operand 3	Operand 4	Instruction	Comments	Line No.
17191								
17192								
17193								
17194	062176	001374				BNE 25\$		
17195	062200	005303				DEC R3	: \$STMP2	
17196	062202	000766				BR 24\$		
17197	062204	005002				CLR R2	: WD.CNT	
17198	062206	017767	133422	133126		MOV @ML.REG+230,PD.TEMP		7892
17199	062214	152777	000020	133302		BISB #20,@ML.REG+120		7894
17200	062222	012703	000001			MOV #1,R3	: *,\$STMP2	7895
17201	062226	001411				BEQ 32\$		7896
17202	062230	016704	117662			MOV LSDLY,R4	: *,\$STMP1	
17203	062234	001404				BEQ 31\$		
17204	062236	005066	000006			CLR 6(SP)	: \$STMP	
17205	062242	005304				DEC R4	: \$STMP1	
17206	062244	001374				BNE 30\$		
17207	062246	005303				DEC R3	: \$STMP2	
17208	062250	000766				BR 29\$		
17209	062252	017767	133316	131114		MOV @ML.REG+170,D1.TEMP		
17210	062260	017767	133320	131110		MOV @ML.REG+200,D2.TEMP		
17211	062266	017767	133272	131104		MOV @ML.REG+160,E2.TEMP		
17212	062274	005004				CLR R4	: NIB.PTR	
17213	062276	010403				MOV R4,R3	: NIB.PTR,*	7899
17214	062300	006203				ASR R3		7902
17215	062302	006203				ASR R3		
17216	062304	006203				ASR R3		
17217	062306	062703	015342			ADD #PD.TEMP,R3		
17218	062312	010346				MOV R3,-(SP)		
17219	062314	010446				MOV R4,-(SP)	: NIB.PTR,*	
17220	062316	042716	177770			BIC #177770,(SP)		
17221	062322	012746	000001			MOV #1,-(SP)		
17222	062326	005046				CLR -(SP)		
17223	062330	004767	121300			JSR PC,BLSGT2		
17224	062334	062706	000010			ADD #10,SP		
17225	062340	005700				TST R0		
17226	062342	001011				BNE 34\$		
17227	062344	010446				MOV R4,-(SP)	: NIB.PTR,*	
17228	062346	010146				MOV R1,-(SP)	: BG.PAT,*	
17229	062350	012746	000012			MOV #12,-(SP)		
17230	062354	060616				ADD SP,(SP)	: ERR.FLG,*	
17231	062356	004767	136014			JSR PC,TST.LNG.WRD		
17232	062362	062706	000006			ADD #6,SP		
17233	062366	026627	000004	000001		CMP 4(SP),#1	: ERR.FLG,*	7906
17234	062374	001001				BNE 35\$		
17235	062376	005216				INC (SP)	: ERR.CNT	
17236	062400	005204				INC R4	: NIB.PTR	7899
17237	062402	020427	000011			CMP R4,#11	: NIB.PTR,*	
17238	062406	003733				BLE 33\$		
17239	062410	005202				INC R2	: WD.CNT	7892
17240	062412	020227	000177			CMP R2,#177	: WD.CNT,*	
17241	062416	003673				BLE 28\$		
17242	062420	021627	000012			CMP (SP),#12	: ERR.CNT,*	7912
17243	062424	003422				BLE 36\$		
17244	062426	104455				TRAP 55		7915
17245	062430	000122				.WORD 122		

```

17247      :ML4AD
17248      :
17249      :
17250 062432 012706      .WORD  ASYNC
17251 062434 026302      .WORD  DUMPER
17252 062436 012746 010630  MOV   #WORD.10,-(SP)
17253 062442 012746 011276  MOV   #WORD.51,-(SP)
17254 062446 012746 011270  MOV   #WORD.50,-(SP)
17255 062452 012746 010256  MOV   #THR.FMT,-(SP)
17256 062456 012746 000004  MOV   #4,-(SP)
17257 062462 010600      MOV   SP,R0
17258 062464 104414      TRAP  14
17259 062466 062706 000012  ADD   #12,SP
17260 062472 005266 000002 36$:  INC   2(SP)
17261 062476 005101      COM   R1
17262 062500 005205      INC   R5
17263 062502 020527 000001  CMP   R5,#1
17264 062506 003002      BGT   37$
17265 062510 000167 177262  JMP   20$
17266 062514 104467      37$:  TRAP  67
17267 062516 006000      ROR   R0
17268 062520 103002      BHS   38$
17269 062522 000167 176434  JMP   1$
17270 062526 062706 000010 38$:  ADD   #10,SP
17271 062532 000207      RTS   PC
17272
17273      : Routine Size: 377 words
17274      : Maximum stack depth per invocation: 18 words
17279
17280
17284
17285
17289 062534      .SBTTL T31 TEST CODE SECTION
17290 062534 004767 176412  T31::
17291 062540 104466      1$:  JSR   PC,$T31
17292 062542 006000      TRAP  66
17293 062544 103773      RCR   R0
17294 062546 000207      BLO   1$
17295      RTS   PC
17296
17297      : Routine Size: 6 words
      : Maximum stack depth per invocation: 0 words
7916
7914
7919
7920
7868
7921
7772
7923

```

17303 :ML4AD
17304 :
17305 :
17306 :

TEST CODE SECTION

7925 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (83)

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (84)

17308 :ML4AD
17309 :
17310 :
17311 :
17312 :
17313 :
17314 :
17315 :
17316 :
17317 :
17318 :
17319 :
17320 :
17321 :
17322 :
17323 :
17324 :
17325 :
17326 :
17327 :
17328 :
17329 :
17330 :
17331 :
17332 :
17333 :
17334 :
17335 :
17336 :
17337 :
17338 :
17339 :
17340 :
17341 :
17342 :
17343 :
17344 :
17345 :
17346 :
17347 :
17348 :
17349 :
17350 :
17351 :
17352 :
17353 :
17354 :
17355 :
17356 :
17357 :
17358 :
17359 :
17360 :
17361 :
17362 :

7926
7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958
7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 32

TEST NAME: ARRAY MODULE SELECTION TEST

TEST DESCRIPTION:

TEST FOR UNIQUE ARRAY MODULE
SELECTION BY:

1. WRITING THE RESPECTIVE ARRAY
MODULE POSITION NUMBER INTO
THE FIRST GOOD NIBBLE FOUND
IN THE ARRAY. DO FOR ALL
PRESENT ARRAYS.
2. READ THE ARRAYS FOR THEIR
RESPECTIVE POSTION NUMBERS.

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

WRDS_TSTED,
ARR_SEL,
FND_GD_NIB,
ARR_NUM;

ARR_SEL = ZEROES;
ARR_SEL = .ARR_SEL - .ARR_INC;

incr ARR_CNT from 0 to .OP_NUM_ARR do

begin
CLR_MBUS;
FND_GD_NIB = ZERO;
WRDS_TSTED = ZERO;
DAT_DM = ONE;
MLWC - not 255;

!WORDS 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

17364 :ML4AD
17365 :
17366 :
17367 :
17368 :
17369 :
17370 :
17371 :
17372 :
17373 :
17374 :
17375 :
17376 :
17377 :
17378 :
17379 :
17380 :
17381 :
17382 :
17383 :
17384 :
17385 :
17386 :
17387 :
17388 :
17389 :
17390 :
17391 :
17392 :
17393 :
17394 :
17395 :
17396 :
17397 :
17398 :
17399 :
17400 :
17401 :
17402 :
17403 :
17404 :
17405 :
17406 :
17407 :
17408 :
17409 :
17410 :
17411 :
17412 :
17413 :
17414 :
17415 :
17416 :
17417 :
17418 :

TEST CODE SECTION

```

MLBA = IO_BUF;
ARR_SEL = .ARR_SEL + .ARR_INC;
MLDA = .ARR_SEL;
MLCS1 = write;

do
  begin
  DELAY (ONE_US);
  PD_TEMP = .MLPD;
  WRDS_TSTED = .WRDS_TSTED + 1;

  incr CNT from 0 to 8 do
    if .PD_TEMP [.CNT] IS_NOT_SET
    then
      begin
        case .CNT from 0 to 8 of
          set
            [0] :
              (MLD1)<0, 4> = .ARR_CNT;
            [1] :
              (MLD1)<4, 4> = .ARR_CNT;
            [2] :
              (MLD1)<8, 4> = .ARR_CNT;
            [3] :
              (MLD1)<12, 4> = .ARR_CNT;
            [4] :
              (MLD2)<0, 4> = .ARR_CNT;
            [5] :
              (MLD2)<4, 4> = .ARR_CNT;
            [6] :
              (MLD2)<8, 4> = .ARR_CNT;
            [7] :
              (MLD2)<12, 4> = .ARR_CNT;
            [8] :
              (MLE2)<8, 4> = .ARR_CNT;
          tes:
        FND_GD_NIB = ONE;
      exitloop;
      end;

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (84)

```

!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

!NIBBLE 0
!NIBBLE 1
!NIBBLE 2
!NIBBLE 3
!NIBBLE 4
!NIBBLE 5
!NIBBLE 6
!NIBBLE 7
!NIBBLE 8

!SET FOUND GOOD NIBBLE FLG
!EXIT THE LOOP

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (84)

```

17420 :ML4AD
17421 :
17422 :
17423 :      8030
17424 :      8031
17425 :      8032
17426 :      8033
17427 :      8034
17428 :      8035
17429 :      8036
17430 :      8037
17431 :      8038
17432 :      8039
17433 :      8040
17434 :      8041
17435 :      8042
17436 :      8043
17437 :      8044
17438 :      8045
17439 :      8046
17440 :      8047
17441 :      8048
17442 :      8049
17443 :      8050
17444 :      8051
17445 :      8052
17446 :      8053
17447 :      8054
17448 :      8055
17449 :      8056
17450 :      8057
17451 :      8058
17452 :      8059
17453 :      8060
17454 :      8061
17455 :      8062
17456 :      8063
17457 :      8064
17458 :      8065
17459 :      8066
17460 :      8067
17461 :      8068
17462 :      8069
17463 :      8070
17464 :      8071
17465 :      8072
17466 :      8073
17467 :      8074
17468 :      8075
17469 :      8076
17470 :      8077
17471 :      8078
17472 :      8079
17473 :      8080
17474 :      8081

TEST CODE SECTION

      DAT_CLK = ONE;
      end
until (.FND_GD_NIB IS_SET ) or (.WRDS_TSTED eql 14);
if .WRDS_TSTED eql 14
then
begin
ERRDF (142, INTER, DUMPER);
PRINTB (SEV_FMT, WRD_14, PHR_10, FNC_15, WRD_12, FNC_17, WRD_37, WRD_56);
EXIT_TST;
end;

end;

ARR_SEL = ZEROES;
ARR_SEL = .ARR_SEL - .ARR_INC;

incr ARR_CNT from 0 to .OP_NUM_ARR do
begin
BGNSUB;
CLR_MBUS;
FND_GD_NIB = ZERO;
DAT_DM = ONE;
MLWC = not 255;
MLBA = IO_BUF;
ARR_SEL = .ARR_SEL + .ARR_INC;
MLDA = .ARR_SEL;
MLCS1 = read;

do
begin
DELAY (ONE_US);
PD_TEMP = .MLPD;

incr CNT from 0 to 8 do
if .PD_TEMP [.CNT] IS_NOT_SET
then
begin
DAT_CLK = ONE;

case .CNT from 0 to 8 of
set
[0] :
ARR_NUM = .(MLD1)<0, 4>;
[1] :
ARR_NUM = .(MLD1)<4, 4>;
[2] :
ARR_NUM = .(MLD1)<8, 4>;

```

```

!CLOCK GOOD NIBBLE INTO MEMORY & GET NXT PROM WRD
!DO UNTIL FOUND GOOD NIBBLE OR 14 WRDS TSTED
!IF 14 WORDS TSTED
!THEN ERROR AND EXIT TESTED
!START ARR SEL BACK ONE AGAIN
!TEST ALL PRESENT ARRAYS
!SET DATA DIAG MODE
!LOAD WORD COUNT
!LOAD UBUS ADDRESS
!LOOK AT NEXT ARRAY
!LOAD DSA WITH ARRAY SELECT
!DO A READ FUNCTION
!THIS LOOP READS ARRAYS FOR ARRAY NUMBERS
!GET THE PROM DATA
!LOOK AT 9 NIBBLES
!FIND THE GOOD NIBBLES WHERE ARR NUM IS STORED
!CLOCK ARRAY WORD OUT
!SELECT AND READ GOOD NIBBLE
!NIBBLE 0
!NIBBLE 1
!NIBBLE 2

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (84)

17476 :ML4AD
17477 :
17478 :
17479 : 8082
17480 : 8083
17481 : 8084
17482 : 8085
17483 : 8086
17484 : 8087
17485 : 8088
17486 : 8089
17487 : 8090
17488 : 8091
17489 : 8092
17490 : 8093
17491 : 8094
17492 : 8095
17493 : 8096
17494 : 8097
17495 : 8098
17496 : 8099
17497 : 8100
17498 : 8101
17499 : 8102
17500 : 8103
17501 : 8104
17502 : 8105
17503 : 8106
17504 : 8107
17505 : 8108
17506 : 8109
17507 : 8110
17508 : 8111
17509 : 8112
17510 : 8113
17511 : 8114
17512 : 8115
17513 : 8116
17514 : 8117
17515 : 8118
17516 : 8119
17517 : 8120
17518 : 8121
17522 :
17523 :
17527 062550 004167 122036
17528 062554 162706 000006
17529 062560 005005
17530 062562 166705 131216

TEST CODE SECTION

```

[3] :
    ARR_NUM = .(MLD1)<12, 4>;      !NIBBLE 3

[4] :
    ARR_NUM = .(MLD2)<0, 4>;      !NIBBLE 4

[5] :
    ARR_NUM = .(MLD2)<4, 4>;      !NIBBLE 5

[6] :
    ARR_NUM = .(MLD2)<8, 4>;      !NIBBLE 6

[7] :
    ARR_NUM = .(MLD2)<12, 4>;     !NIBBLE 7

[8] :
    ARR_NUM = .(MLE2)<8, 4>;     !NIBBLE 8
tes;

    FND_GD_NIB = ONE;           !SET FND GD NIB FLG
    exit loop;                  !EXIT LOOP
end;

    DAT_CLK = ONE;             !CLOCK OUT NEXT PROM LOCATION
end
until .FND_GD_NIB IS_SET;     !REPEAT UNTIL FOUND THE GOOD NIBBLE

if .ARR_CNT neq .ARR_NUM      !SEE IF ARRAY CONTAINS IT'S ARR NUM
then
begin                          !ERROR IF NOT THERE
ERRDF (83, ASYNC, DUMPER);
PRINTB (FOR_FMT, WRD_39, FNC 17, WRD_37, WRD_10);
PRINTB (FMT_14, .ARR_CNT, .ARR_NUM);
end;

ENDSUB;
end;

ENDTST;

```

```

$T32: .SBTTL $T32 TEST CODE SECTION
      JSR R1,$SAVE5
      SUB #6,SP
      CLR R5
      SUB ARR.INC,R5
      :
      : ARR.SEL
      : *,ARR.SEL

```

7924
7968
7969

Address	OpCode	Operand	Comment	Code Section	Register	Value	Count
17532							
17533				TEST CODE SECTION			
17534							
17535	062566	016766	131210	000002	MOV	OP.NUM.ARR,2(SP)	
17536	062574	005003			CLR	R3	ARR.CNT
17537	062576	000167	000702		JMP	27\$	
17538	062602	152777	000040	132634	1\$: BISB	#40,@ML.REG+40	
17539	062610	016702	133216		MOV	ML.DUT,R2	
17540	062614	042702	177770		BIC	#177770,R2	
17541	062620	142777	000007	132616	BICB	#7,@ML.REG+40	
17542	062626	150277	132612		BISB	R2,@ML.REG+40	
17543	062632	005016			CLR	(SP)	FND.GD.NIB
17544	062634	005004			CLR	R4	WRDS.TSTED
17545	062636	152777	000010	132660	BISB	#10,@ML.REG+120	
17546	062644	012777	177400	132542	MOV	#-400,@ML.REG+10	
17547	062652	012777	014022	132544	MOV	#10.BUF,@ML.REG+20	
17548	062660	066705	131120		ADD	ARR.INC,R5	*,ARR.SEL
17549	062664	010577	132544		MOV	R5,@ML.REG+30	ARR.SEL,*
17550	062670	012777	000061	132506	MOV	#61,@ML.REG	
17551	062676	012701	000001		MOV	#1,R1	*,\$\$TMP2
17552	062702	001411			BEQ	6\$	*,\$\$TMP2
17553	062704	016702	117206		MOV	LSPLY,R2	*,\$\$TMP1
17554	062710	001404			BEQ	5\$	
17555	062712	005066	000004		CLR	4(SP)	\$\$TMP
17556	062716	005302			DEC	R2	\$\$TMP1
17557	062720	001374			BNE	4\$	
17558	062722	005301			DEC	R1	\$\$TMP2
17559	062724	000766			BR	3\$	
17560	062726	017767	132702	132406	6\$: MOV	@ML.REG+230,PD.TEMP	
17561	062734	005204			INC	R4	WRDS.TSTED
17562	062736	005002			CLR	R2	CNT
17563	062740	010201			7\$: MOV	R2,R1	CNT,*
17564	062742	006201			ASR	R1	
17565	062744	006201			ASR	R1	
17566	062746	006201			ASR	R1	
17567	062750	062701	015342		ADD	#PD.TEMP,R1	
17568	062754	010146			MOV	R1,-(SP)	
17569	062756	010246			MOV	R2,-(SP)	CNT,*
17570	062760	042716	177770		BIC	#177770,(SP)	
17571	062764	012746	000001		MOV	#1,-(SP)	
17572	062770	005046			CLR	-(SP)	
17573	062772	004767	120636		JSR	PC,BLSGT2	
17574	062776	062706	000010		ADD	#10,SP	
17575	063002	005700			TST	R0	
17576	063004	001155			BNE	23\$	
17577	063006	010201			MOV	R2,R1	CNT,*
17578	063010	006301			ASL	R1	
17579	063012	066107	063016		8\$: ADD	8\$(R1),PC	
17580	063016	000022			8\$: .WORD	9\$-8\$	
17581	063020	000040			8\$: .WORD	10\$-8\$	
17582	063022	000072			8\$: .WORD	12\$-8\$	
17583	063024	000112			8\$: .WORD	13\$-8\$	
17584	063026	000146			8\$: .WORD	15\$-8\$	
17585	063030	000164			8\$: .WORD	16\$-8\$	
17586	063032	000216			8\$: .WORD	18\$-8\$	

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

7971

7972

7974

7975

7976

7977

7978

7979

7980

7981

7985

7986

7987

7989

7991

7995

17588			:ML4AD					
17589			:			TEST CODE SECTION		
17590								
17591	063034	000236				.WORD 19\$-8\$		
17592	063036	000272				.WORD 21\$-8\$		
17593	063040	010301				MOV R3,R1		
17594	063042	042701	177760		9\$:	BIC #177760,R1	: ARR.CNT,*	7999
17595	063046	142777	000017	132520		BICB #17,@ML.REG+170		
17596	063054	000412				BR 11\$		
17597	063056	010301			10\$:	MOV R3,R1	: ARR.CNT,*	8002
17598	063060	006301				ASL R1		
17599	063062	006301				ASL R1		
17600	063064	006301				ASL R1		
17601	063066	006301				ASL R1		
17602	063070	042701	177417			BIC #177417,R1		
17603	063074	142777	000360	132472		BICB #360,@ML.REG+170		
17604	063102	150177	132466		11\$:	BISB R1,@ML.REG+170		
17605	063106	000511				BR 22\$		
17606	063110	010301			12\$:	MOV R3,R1	: ARR.CNT,*	7995
17607	063112	000301				SWAB R1		8005
17608	063114	042701	170377			BIC #170377,R1		
17609	063120	042777	007400	132446		BIC #7400,@ML.REG+170		
17610	063126	000413				BR 14\$		
17611	063130	010301			13\$:	MOV R3,R1	: ARR.CNT,*	8008
17612	063132	000301				SWAB R1		
17613	063134	006301				ASL R1		
17614	063136	006301				ASL R1		
17615	063140	006301				ASL R1		
17616	063142	006301				ASL R1		
17617	063144	042701	007777			BIC #7777,R1		
17618	063150	042777	170000	132416		BIC #170000,@ML.REG+170		
17619	063156	050177	132412		14\$:	BIS R1,@ML.REG+170		
17620	063162	000463				BR 22\$		
17621	063164	010301			15\$:	MOV R3,R1	: ARR.CNT,*	7995
17622	063166	042701	177760			BIC #177760,R1		8011
17623	063172	142777	000017	132404		BICB #17,@ML.REG+200		
17624	063200	000412				BR 17\$		
17625	063202	010301			16\$:	MOV R3,R1	: ARR.CNT,*	8014
17626	063204	006301				ASL R1		
17627	063206	006301				ASL R1		
17628	063210	006301				ASL R1		
17629	063212	006301				ASL R1		
17630	063214	042701	177417			BIC #177417,R1		
17631	063220	142777	000360	132356		BICB #360,@ML.REG+200		
17632	063226	150177	132352		17\$:	BISB R1,@ML.REG+200		
17633	063232	000437				BR 22\$		
17634	063234	010301			18\$:	MOV R3,R1	: ARR.CNT,*	7995
17635	063236	000301				SWAB R1		8017
17636	063240	042701	170377			BIC #170377,R1		
17637	063244	042777	007400	132332		BIC #7400,@ML.REG+200		
17638	063252	000413				BR 20\$		
17639	063254	010301			19\$:	MOV R3,R1	: ARR.CNT,*	8020
17640	063256	000301				SWAB R1		
17641	063260	006301				ASL R1		
17642	063262	006301				ASL R1		

Address	Op Code	Operand 1	Operand 2	Operand 3	Label	Instruction	Comments	Line Number
17644					:ML4AD			
17645					:			
17646					:	TEST CODE SECTION		
17647	063264	006301				ASL R1		
17648	063266	006301				ASL R1		
17649	063270	042701	007777			BIC #7777,R1		
17650	063274	042777	170000	132302		BIC #170000,@AML.REG+200		
17651	063302	050177	132276		20\$:	BIS R1,@AML.REG+200		
17652	063306	000411				BR 22\$		
17653	063310	010301			21\$:	MOV R3,R1	: ARR.CNT,*	7995
17654	063312	000301				SWAB R1		8023
17655	063314	042701	170377			BIC #170377,R1		
17656	063320	042777	007400	132236		BIC #7400,@AML.REG+160		
17657	063326	050177	132232			BIS R1,@AML.REG+160		
17658	063332	012716	000001		22\$:	MOV #1,(SP)	: *,FND.GD.NIB	8026
17659	063336	000406				BR 24\$		8027
17660	063340	005202			23\$:	INC R2	: CNT	7989
17661	063342	020227	000010			CMP R2,#10	: CNT,*	
17662	063346	003002				BGT 24\$		
17663	063350	000167	177364			JMP 7\$		
17664	063354	152777	000020	132142	24\$:	BISB #20,@AML.REG+120		
17665	063362	021627	000001			CMP (SP),#1	: FND.GD.NIB,*	8030
17666	063366	001405				BEQ 25\$		8032
17667	063370	020427	000016			CMP R4,#16	: WRDS.TSTED,*	
17668	063374	001402				BEQ 25\$		
17669	063376	000167	177274			JMP 2\$		
17670	063402	020427	000016		25\$:	CMP R4,#16	: WRDS.TSTED,*	8034
17671	063406	001035				BNE 26\$		8037
17672	063410	104455				TRAP 55	:	
17673	063412	000216				.WORD 216		8038
17674	063414	013114				.WORD INTER		
17675	063416	026302				.WORD DUMPER		
17676	063420	012746	011344			MOV #WRD.56,-(SP)		
17677	063424	012746	011122			MOV #WRD.37,-(SP)		
17678	063430	012746	012404			MOV #FNC.17,-(SP)		
17679	063434	012746	010650			MOV #WRD.12,-(SP)		
17680	063440	012746	012360			MOV #FNC.15,-(SP)		
17681	063444	012746	012006			MOV #PHR.10,-(SP)		
17682	063450	012746	010664			MOV #WRD.14,-(SP)		
17683	063454	012746	010342			MOV #SEV.FMT,-(SP)		
17684	063460	012746	000010			MOV #10,-(SP)		
17685	063464	010600				MOV SP,R0	: SP,*	
17686	063466	104414				TRAP 14		
17687	063470	104463				TRAP 63		
17688	063472	062706	000022			ADD #22,SP		
17689	063476	000167	000562			JMP 56\$		8034
17690	063502	005203			26\$:	INC R3	: ARR.CNT	8036
17691	063504	020366	000002		27\$:	CMP R3,2(SP)	: ARR.CNT,*	7971
17692	063510	003002				BGT 28\$		
17693	063512	000167	177064			JMP 1\$		
17694	063516	005005			28\$:	CLR R5	: ARR.SEL	8044
17695	063520	166705	130260			SUB ARR.INC,R5	: *,ARR.SEL	8045
17696	063524	016766	130252	000002		MOV OP.NUM.ARR,2(SP)		8047
17697	063532	005004				CLR R4	: ARR.CNT	
17698	063534	000167	000516			JMP 55\$		

Address	Label	Instruction	Comment	Page	Time	Code	Page	
17700			:ML4AD					
17701			:		29-Mar-1982 16:23:04	TOPS		
17702			:		29-Mar-1982 16:21:03	PA:<		
17703	063540	104402	29\$: TRAP	2				
17704	063542	152777	BISB	#40,@ML.REG+40	:		8048	
17705	063550	016702	MOV	ML.DUT,R2	:		8049	
17706	063554	042702	BIC	#177770,R2	:			
17707	063560	142777	BICB	#7,@ML.REG+40				
17708	063566	150277	BISB	R2,@ML.REG+40				
17709	063572	005016	CLR	(SP)	:	FND.GD.NIB		
17710	063574	152777	BISB	#10,@ML.REG+120	:		8051	
17711	063602	012777	MOV	#-400,@ML.REG+10	:		8052	
17712	063610	012777	MOV	#10.BUF,@ML.REG+20	:		8053	
17713	063616	066705	ADD	ARR.INC,R5	:	*.ARR.SEL	8054	
17714	063622	010577	MOV	R5,@ML.REG+30	:	ARR.SEL,*	8055	
17715	063626	012777	MOV	#71,@ML.REG	:		8056	
17716	063634	012701	30\$: MOV	#1,R1	:	*.SSTMP2	8057	
17717	063640	001411	31\$: BEQ	34\$:		8061	
17718	063642	016702	MOV	LSPLY,R2	:	*.SSTMP1		
17719	063646	001404	BEQ	33\$:			
17720	063650	005066	32\$: CLR	4(SP)	:	SSTMP		
17721	063654	005302	DEC	R2	:	SSTMP1		
17722	063656	001374	BNE	32\$:			
17723	063660	005301	33\$: DEC	R1	:	SSTMP2		
17724	063662	000766	BR	31\$:			
17725	063664	017767	34\$: MOV	@ML.REG+230,PD.TEMP	:		8062	
17726	063672	005002	CLR	R2	:	CNT	8064	
17727	063674	010201	35\$: MOV	R2,R1	:	CNT,*	8066	
17728	063676	006201	ASR	R1	:			
17729	063700	006201	ASR	R1	:			
17730	063702	006201	ASR	R1	:			
17731	063704	062701	015342	ADD	#PD.TEMP,R1			
17732	063710	010146	MOV	R1, -(SP)				
17733	063712	010246	MOV	R2, -(SP)	:	CNT,*		
17734	063714	042716	177770	BIC	#177770,(SP)			
17735	063720	012746	000001	MOV	#1, -(SP)			
17736	063724	005046	CLR	-(SP)				
17737	063726	004767	117702	JSR	PC,BLSGT2			
17738	063732	062706	000010	ADD	#10,SP			
17739	063736	005700	TST	R0				
17740	063740	001070	BNE	50\$				
17741	063742	152777	000020	131554	BISB	#20,@ML.REG+120		
17742	063750	010201	MOV	R2,R1	:	CNT,*	8069	
17743	063752	006301	ASL	R1	:		8071	
17744	063754	066107	063760	ADD	36\$(R1),PC			
17745	063760	000022	36\$: .WORD	37\$-36\$				
17746	063762	000030	.WORD	38\$-36\$				
17747	063764	000036	.WORD	39\$-36\$				
17748	063766	000044	.WORD	40\$-36\$				
17749	063770	000052	.WORD	41\$-36\$				
17750	063772	000060	.WORD	42\$-36\$				
17751	063774	000076	.WORD	44\$-36\$				
17752	063776	000104	.WORD	45\$-36\$				
17753	064000	000122	.WORD	47\$-36\$				
17754	064002	117703	131566	37\$: MOVB	@ML.REG+170,R3	:	*.ARR.NUM	8075

Address	Op Code	Op Data	Op Data	Label	Instruction	Comments	Address	
17756								
17757								
17758								
17759	064006	000440			BR 49\$			
17760	064010	117703	131560	38\$:	MOVB @ML.REG+170,R3	: *,ARR.NUM	8078	
17761	064014	000413			BR 43\$			
17762	064016	017703	131552	39\$:	MOV @ML.REG+170,R3	: *,ARR.NUM	8081	
17763	064022	000431			BR 48\$		8081	
17764	064024	017703	131544	40\$:	MOV @ML.REG+170,R3	: *,ARR.NUM	8084	
17765	064030	000417			BR 46\$		8084	
17766	064032	117703	131546	41\$:	MOVB @ML.REG+200,R3	: *,ARR.NUM	8087	
17767	064036	000424			BR 49\$		8087	
17768	064040	117703	131540	42\$:	MOVB @ML.REG+200,R3	: *,ARR.NUM	8090	
17769	064044	006203		43\$:	ASR R3	: ARR.NUM	8090	
17770	064046	006203			ASR R3	: ARR.NUM		
17771	064050	006203			ASR R3	: ARR.NUM		
17772	064052	006203			ASR R3	: ARR.NUM		
17773	064054	000415			BR 49\$: ARR.NUM		
17774	064056	017703	131522	44\$:	MOV @ML.REG+200,R3	: *,ARR.NUM	8093	
17775	064062	000411			BR 48\$		8093	
17776	064064	017703	131514	45\$:	MOV @ML.REG+200,R3	: *,ARR.NUM	8096	
17777	064070	006203		46\$:	ASR R3	: ARR.NUM	8096	
17778	064072	006203			ASR R3	: ARR.NUM		
17779	064074	006203			ASR R3	: ARR.NUM		
17780	064076	006203			ASR R3	: ARR.NUM		
17781	064100	000402			BR 48\$			
17782	064102	017703	131456	47\$:	MOV @ML.REG+160,R3	: *,ARR.NUM	8099	
17783	064106	000303		48\$:	SWAB R3	: ARR.NUM	8099	
17784	064110	042703	177760	49\$:	BIC #177760,R3	: *,ARR.NUM		
17785	064114	012716	000001		MOV #1,(SP)	: *,FND.GD.NIB	8102	
17786	064120	000404			BR 51\$		8103	
17787	064122	005202		50\$:	INC R2	: CNT	8064	
17788	064124	020227	000010		CMP R2,#10	: CNT,*		
17789	064130	003661			BLE 35\$			
17790	064132	152777	000020	131364	51\$:	BISB #20,@ML.REG+120	: FND.GD.NIB,*	8106
17791	064140	021627	000001		CMP (SP),#1	: FND.GD.NIB,*	8108	
17792	064144	001233			BNE 30\$			
17793	064146	020403			CMP R4,R3	: ARR.CNT,ARR.NUM	8110	
17794	064150	001434			BEQ 52\$			
17795	064152	104455			TRAP 55			
17796	064154	000123			.WORD 123		8113	
17797	064156	012706			.WORD ASYNC			
17798	064160	026302			.WORD DUMPER			
17799	064162	012746	010630		MOV #WRD.10,-(SP)			
17800	064166	012746	011122		MOV #WRD.37,-(SP)		8114	
17801	064172	012746	012404		MOV #FNC.17,-(SP)			
17802	064176	012746	011140		MOV #WRD.39,-(SP)			
17803	064202	012746	010270		MOV #FOR.FMT,-(SP)			
17804	064206	012746	000005		MOV #5,-(SP)			
17805	064212	010600			MOV SP,R0	: SP,*		
17806	064214	104414			TRAP 14			
17807	064216	010316			MOV R3,(SP)	: ARR.NUM,*	8115	
17808	064220	010446			MOV R4,-(SP)	: ARR.CNT,*		
17809	064222	012746	007330		MOV #FMT.14,-(SP)			
17810	064226	012746	000003		MOV #3,-(SP)			

17812
17813
17814
17815 064232 010600
17816 064234 104414
17817 064236 062706 000022
17818 064242 104467
17819 064244 006000
17820 064246 103002
17821 064250 000167 177264
17822 064254 005204
17823 064256 020466 000002
17824 064262 003772
17825 064264 062706 000006
17826 064270 000207

:ML4AD
:
TEST CODE SECTION
:
MOV SP,R0
TRAP 14 ; SP,*
ADD #22,SP
52\$: TRAP 67 ;
ROR R0 ;
BHIS 54\$
53\$: JMP 29\$
54\$: INC R4 ; ARR.CNT
55\$: CMP R4,2(SP) ; ARR.CNT,*
BLE 53\$
56\$: ADD #6,SP ;
RTS PC

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

8112
8116

8047

7924

: Routine Size: 425 words
: Maximum stack depth per invocation: 18 words

17827
17828
17829
17834
17835
17839
17840
17844 064272
17845 064272 004767 176252
17846 064276 104466
17847 064300 006000
17848 064302 103773
17849 064304 000207

.SBTTL T32 TEST CODE SECTION
T32::
1\$: JSR PC,\$T32 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

8119

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

17850
17851
17852
17857
17858
17859 ; 8122 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (85)

17861 :ML4AD
17862 :
17863 :
17864 :
17865 :
17866 :
17867 :
17868 :
17869 :
17870 :
17871 :
17872 :
17873 :
17874 :
17875 :
17876 :
17877 :
17878 :
17879 :
17880 :
17881 :
17882 :
17883 :
17884 :
17885 :
17886 :
17887 :
17888 :
17889 :
17890 :
17891 :
17892 :
17893 :
17894 :
17895 :
17896 :
17897 :
17898 :
17899 :
17900 :
17901 :
17902 :
17903 :
17904 :
17905 :
17906 :
17907 :
17908 :
17909 :
17910 :
17911 :
17912 :
17913 :
17914 :
17915 :

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
8170
8171
8172
8173
8174

TEST CODE SECTION

! BGNTST;

! ++

TEST NUMBER: TST 33

TEST NAME: SEQUENCER EXISTENCE TEST

TEST DESCRIPTION:

TEST SEQUENCER TIMING AND CONTROL
LOGIC FOR EXISTENCE BY:

1. FIRST WRITING THE GOOD BLOCK VIA DAT_DM MODE WITH A BACKGROUND PATTERN OF ONES.
2. VIA A MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
3. THEN VIA DAT_DM READ GOOD NIBBLES IN THE GOOD BLOCK AND XOR THEM AGAINST THE BACKGROUND PATTERN.

RECORD THE NUMBER OF NIBBLES THAT WERE UNCHANGED OR PARTIALLY CHANGED BY THE MBUS WRITE.
4. REPORT A FATAL ERROR AND DROP THIS UNIT IF THE NIBBLES TESTED EQUALS THE COUNT OF UNCHANGED NIBBLES.

REPORT AN INTERMEDIATE DIAGNOSTIC MESSAGE IF AT LEAST SOME NIBBLES WERE CHANGED BY THE MBUS WRITE.

IMPLICIT INPUTS:

PD TEMP:

A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND ACCESSED FROM.

IO BUF:

A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTIONS ARE FOUND.

Local

BG_PAT,
SUM_BAD,
NIB_TSTED,
ALL_BAD,
RESULT;

!BACK GROUND PATTERN
!SUM NIBBLE ARE BAD
!NIBBLES TESTED
!ALL NIBBLES ARE BAD
!RESULTS OF XOR

CLR MBUS:

BG_PAT = ONES;
MLD1 = .BG_PAT;
MLD2 = .BG_PAT;
MLE2 = .BG_PAT;

!BACKGROUND EQL ONES
!LOAD DATA DIAG REG WITH BG PAT

```

17917 :ML4AD
17918 :
17919 :
17920 :      8175 DAT_DM_XFER ();
17921 :      8176 MLC51 = write;
17922 :      8177
17923 :      8178 incr WD_CNT from 0 to 127 do
17924 :      8179   begin
17925 :      8180     DELAY (ONE_US);
17926 :      8181     DAT_CLK = ONE;
17927 :      8182   end;
17928 :      8183
17929 :      8184 CLR_MBUS;
17930 :      8185 IO_BUF = ZEROES;
17931 :      8186 BAI = ONE;
17932 :      8187 GD_BLK_XFER ();
17933 :      8188 MLC51 = write;
17934 :      8189 TIME_OUT_LOOP;
17935 :      8190 BGNS03;
17936 :      8191 CLR_MBUS;
17937 :      8192 ALL_BAD = ZEROES;
17938 :      8193 SUM_BAD = ZEROES;
17939 :      8194 NIB_TSTED = ZEROES;
17940 :      8195 DAT_DM_XFER ();
17941 :      8196 MLC51 = read;
17942 :      8197 DELAY (ONE_US);
17943 :      8198
17944 :      8199 incr WD_CNT from 0 to 112 do
17945 :      8200   begin
17946 :      8201     PD_TEMP = .MLPD;
17947 :      8202     DAT_CLK = ONE;
17948 :      8203     DELAY (ONE_US);
17949 :      8204     RD_LNG_WRD;
17950 :      8205
17951 :      8206     incr NIB_PTR from 0 to 8 do
17952 :      8207       begin
17953 :      8208         if .PD_TEMP [.NIB_PTR] eql ZERO
17954 :      8209         then
17955 :      8210           begin
17956 :      8211             NIB_TSTED = .NIB_TSTED + 1;
17957 :      8212             XOR_LNG_WRD (.NIB_PTR, .BG_PAT, RESULT);
17958 :      8213           end
17959 :      8214           if .RESULT<0, 4> eql ZERO
17960 :      8215           then
17961 :      8216             ALL_BAD = .ALL_BAD + 1
17962 :      8217           else
17963 :      8218             if .RESULT<0, 4> neq %0'17' then SUM_BAD = .SUM_BAD + 1;
17964 :      8219           end
17965 :      8220         end;
17966 :      8221       end;
17967 :      8222     end;
17968 :      8223   end;
17969 :      8224 end;
17970 :      8225
17971 :      8226 ENDSUB;

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (85)

!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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI 4 (85)

```

17973 :ML4AD
17974 :
17975 :
17976 :      8227
17977 :      8228 if .NIB_TSTED eql .ALL_BAD
17978 :      8229 then
17979 :      8230 begin
17980 :      8231 ERRDF (84, SYNC, DUMPER);
17981 :      8232 PRINTB (SEV_FMT, WRD_22, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
17982 :      8233 DODU (.ML_LDN);
17983 :      8234 DOCLN;
17984 :      8235 end
17985 :      8236 else
17986 :      8237
17987 :      8238 if .SUM_BAD gtr ZERO
17988 :      8239 then
17989 :      8240 begin
17990 :      8241 ERRDF (85, INTER, DUMPER);
17991 :      8242 PRINTB (SIX_FMT, PHR_4, WRD_9, WRD_12, WRD_23, FNC_5, WRD_19);
17992 :      8243 end;
17993 :      8244
17994 :      8245 ENDTST;
17998 :
17999 :

```

!WHERE ALL NIBBLES XFERR'ED ALL BAD?
!ERROR IF ALL BAD

!SEE IF SEE NIBBLE WERE BAD?

!SOME BAD IS OK SO GIVE INTERMEDIATE ERROR

Address	Hex	Dec	Label	Instruction	Comment	Address
18003	064306	004167	120300	.SBTTL		
18004	064312	162706	000010	JSR	R1,\$\$SAVE5	
18005	064316	152777	000040	SUB	#10,SP	8121
18006	064324	016705	131502	BISB	#40,@ML.REG+40	
18007	064330	042705	177770	MOV	ML,DUT,R5	8168
18008	064334	142777	000007	BIC	#177770,R5	
18009	064342	150577	131076	BICB	#7,@ML.REG+40	
18010	064346	012766	177777	BISB	R5,@ML.REG+40	
18011	064354	012777	177777	MOV	#-1,2(SP)	*.BG.PAT
18012	064362	016677	000002	MOV	#-1,@ML.REG+170	8171
18013	064370	016677	000002	MOV	2(SP),@ML.REG+200	8172
18014	064376	004767	132010	MOV	2(SP),@ML.REG+160	8173
18015	064402	012777	000061	JSR	PC,DAT.DM.XFER	8174
18016	064410	005000		MOV	#61,@ML.REG	8175
18017	064412	012701	000001	CLR	R0	8176
18018	064416	001411		MOV	#1,R1	8178
18019	064420	016702	115472	BEQ	5\$	*,\$\$TMP2
18020	064424	001404		MOV	LSDLY,R2	
18021	064426	005066	000006	BEQ	4\$	*,\$\$TMP1
18022	064432	005302		CLR	6(SP)	
18023	064434	001374		DEC	R2	\$\$TMP
18024	064436	005301		BNE	3\$	\$\$TMP1
18025	064440	000766		DEC	R1	
18026	064442	152777	000020	BR	2\$	\$\$TMP2
18027	064450	005200	131054	BISB	#20,@ML.REG+120	8181
				INC	R0	8178

Address	Hex	Hex	Hex	Instruction	Comment	Address
18085						
18086						
18087						
18088	064744	006202		ASR R2		
18089	064746	062702	015342	ADD #PD.TEMP,R2		
18090	064752	010246		MOV R2,-(SP)		
18091	064754	010146		MOV R1,-(SP)		
18092	064756	042716	177770	BIC #177770,(SP)	: NIB.PTR,*	
18093	064762	012746	000001	MOV #1,-(SP)		
18094	064766	005046		CLR -(SP)		
18095	064770	004767	116640	JSR PC,BLSGT2		
18096	064774	062706	000010	ADD #10,SP		
18097	065000	005700		TST R0		
18098	065002	001027		BNE 20\$		
18099	065004	005205		INC R5	: NIB.TSTED	8211
18100	065006	010146		MOV R1,-(SP)	: NIB.PTR,*	8212
18101	065010	016646	000004	MOV 4(SP),-(SP)	: BG.PAT,*	
18102	065014	012746	000012	MOV #12,-(SP)		
18103	065020	060616		ADD SP,(SP)	: RESULT,*	
18104	065022	004767	134044	JSR PC,XOR.LNG.WRD		
18105	065026	016602	000012	MOV 12(SP),R2	: RESULT,*	8214
18106	065032	042702	177760	BIC #177760,R2		
18107	065036	001002		BNE 18\$		
18108	065040	005204		INC R4	: ALL.BAD	8216
18109	065042	000405		BR 19\$:	8214
18110	065044	020227	000017	CMP R2,#17	:	8219
18111	065050	001402		BEQ 19\$:	
18112	065052	005266	000006	INC 6(SP)	: SUM.BAD	
18113	065056	062706	000006	ADD #6,SP	:	
18114	065062	005201		INC R1	: NIB.PTR	8210
18115	065064	020127	000010	CMP R1,#10	: NIB.PTR,*	8206
18116	065070	003722		BLE 17\$		
18117	065072	005203		INC R3	: WD.CNT	8199
18118	065074	020327	000160	CMP R3,#160	: WD.CNT,*	
18119	065100	003662		BLE 12\$		
18120	065102	104467		TRAP 67	:	8224
18121	065104	006000		ROR R0		
18122	065106	103615		BLO 7\$		
18123	065110	020504		CMP R5,R4	: NIB.TSTED,ALL.BAD	8228
18124	065112	001037		BNE 21\$		
18125	065114	104455		TRAP 55	:	8231
18126	065116	000124		.WORD 124		
18127	065120	012750		.WORD SYNC		
18128	065122	026302		.WORD DUMPER		
18129	065124	012746	010730	MOV #WRD.19,-(SP)	:	8232
18130	065130	012746	012216	MOV #FNC.5,-(SP)		
18131	065134	012746	010766	MOV #WRD.23,-(SP)		
18132	065140	012746	010650	MOV #WRD.12,-(SP)		
18133	065144	012746	010616	MOV #WRD.9,-(SP)		
18134	065150	012746	011676	MOV #PHR.4,-(SP)		
18135	065154	012746	010752	MOV #WRD.22,-(SP)		
18136	065160	012746	010342	MOV #SEV.FMT,-(SP)		
18137	065164	012746	000010	MOV #10,-(SP)		
18138	065170	010600		MOV SP,R0	: SP,*	
18139	065172	104414		TRAP 14		

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

18141
 18142
 18143
 18144 065174 016700 130630
 18145 065200 104451
 18146 065202 104444
 18147 065204 062706 000022
 18148 065210 000432
 18149 065212 005716
 18150 065214 003430
 18151 065216 104455
 18152 065220 000125
 18153 065222 013114
 18154 065224 026302
 18155 065226 012746 010730
 18156 065232 012746 012216
 18157 065236 012746 010766
 18158 065242 012746 010650
 18159 065246 012746 010616
 18160 065252 012746 011676
 18161 065256 012746 010322
 18162 065262 012746 000007
 18163 065266 010600
 18164 065270 104414
 18165 065272 062706 000020
 18166 065276 062706 000010
 18167 065302 000207
 18168
 18169
 18170
 18175
 18176
 18180
 18181
 18185 065304
 18186 065304 004767 176776
 18187 065310 104466
 18188 065312 006000
 18189 065314 103773
 18190 065316 000207
 18191
 18192
 18193

:ML4AD
 :
 TEST CODE SECTION

```

MOV ML,LUN,RO
TRAP 51
TRAP 44
ADD #22,SP
BR 22$
21$: TST (SP)
BLE 22$
TRAP 55
.WORD 125
.WORD INTER
.WORD DUMPER
MOV #WRD.19,-(SP)
MOV #FNC.5,-(SP)
MOV #WRD.23,-(SP)
MOV #WRD.12,-(SP)
MOV #WRD.9,-(SP)
MOV #PHR.4,-(SP)
MOV #SIX.FMT,-(SP)
MOV #7,-(SP)
MOV SP,RO
TRAP 14
22$: ADD #20,SP
ADD #10,SP
RTS PC

```

: Routine Size: 255 words
 : Maximum stack depth per invocation: 19 words

.SBTTL T33 TEST CODE SECTION

```

T33::
1$: JSR PC,$T33
TRAP 66
ROR R0
BLO 1$
RTS PC

```

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

8233

8230
 8228
 8238

8241

8242

8240
 8121

8243

18202
18203
18204 ; 8246 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (86)

18206 :ML4AD
 18207 :
 18208 :
 18209 :
 18210 :
 18211 :
 18212 :
 18213 :
 18214 :
 18215 :
 18216 :
 18217 :
 18218 :
 18219 :
 18220 :
 18221 :
 18222 :
 18223 :
 18224 :
 18225 :
 18226 :
 18227 :
 18228 :
 18229 :
 18230 :
 18231 :
 18232 :
 18233 :
 18234 :
 18235 :
 18236 :
 18237 :
 18238 :
 18239 :
 18240 :
 18241 :
 18242 :
 18243 :
 18244 :
 18245 :
 18246 :
 18247 :
 18248 :
 18249 :
 18250 :
 18251 :
 18252 :
 18253 :
 18254 :
 18255 :
 18256 :
 18257 :
 18258 :
 18259 :
 18260 :

TEST CODE SECTION

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 DAT_DM MODE WRITE A
BACKGROUND PATTERN OF ONES
INTO THE GOOD BLOCK.
2. VIA MBUS WRITE FUNCTION
WRITE A ZEROES PATTERN INTO
THE GOOD BLOCK.
3. VIA DAT_DM MODE READ GOOD
NIBBLES IN THE GOOD BLOCK FOR
ZEROES.
4. REPEAT WITH COMPLIMENT
DATA AND BACKGROUND PATTERNS.

IMPLICIT INPUTS:

PD_TEMP

A BIT VECTOR OF 16 BITS WHERE
THE READ PROM DATA IS STORED
AND ACCESSED FROM.

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 XO!
!TEST PATTERN

18262	:ML4AD			29-Mar-1982 16:23:04	TOPS-20 Bliss-16 V2(212)
18263	:	TEST CODE SECTION		29-Mar-1982 16:21:03	PA:<NEALE>ML4AD.BLI.4 (86)
18264	:				
18265	:	8299			
18266	:	8300	CLR THRESHOLD;	!CLEAR ERROR PRINT THRESHOLD	
18267	:	8301	DODD FLG = ZERO;		
18268	:	8302	BG PAT = ZEROES;	!BG PAT EQL 0'S	
18269	:	8303	TST_PAT = ONES;	!TST PAT EQL 1'S	
18270	:	8304			
18271	:	8305	incr TWICE from 0 to 1 do	!REPEAT LOOP TWICE	
18272	:	8306	begin		
18273	:	8307	BGNSUB;		
18274	:	8308	CLR MBUS;		
18275	:	8309	MLDT = .BG_PAT;	!LOAD DATA DIAG REG WITH BG PAT	
18276	:	8310	MLD2 = .BG_PAT;		
18277	:	8311	MLE2 = .BG_PAT;		
18278	:	8312	DAT_DM_XFER ();	!SET UP A DATA DIAG XFERR AT THE GOOD BLK	
18279	:	8313	MLCS1 = write;	!DO A WRITE FUNCTION	
18280	:	8314			
18281	:	8315	incr WD_CNT from 0 to 127 do	!LOAD BLOCK WITH BG PAT	
18282	:	8316	begin		
18283	:	8317	DELAY (ONE_US);		
18284	:	8318	DAT_CLK = ONE;		
18285	:	8319	end;		
18286	:	8320			
18287	:	8321	CLR MBUS;		
18288	:	8322	BAI = ONE;	!SET ON FIRST IO BUF ADRS	
18289	:	8323	IO_BUF = .TST_PAT;	!FIRST IO_BUF WORD EQL'S TST_PAT	
18290	:	8324	GD_BLK_XFER ();	!SET UP A GOOD BLOCK XFERR	
18291	:	8325	MLCS1 = write;	!DO A WRITE FUNCTION	
18292	:	8326	TIME_OUT_LOOP;		
18293	:	8327	CLR MBUS;		
18294	:	8328	DAT_DM_XFER ();	!SET UP A DATA DIAG XFERR AT SAME BLOCK	
18295	:	8329	MLCS1 = read;	!DO A READ FUNCTION	
18296	:	8330	DELAY (ONE_US);		
18297	:	8331			
18298	:	8332	incr WD_CNT from 0 to 112 do	!READ 113 LONG WORDS	
18299	:	8333	begin		
18300	:	8334	PD_TEMP = .MLPD;	!GET THE PROM DATA	
18301	:	8335	DAT_CLK = ONE;	!CLOCK OUT THE DATA WORD	
18302	:	8336	DELAY (ONE_US);		
18303	:	8337	RD_LNG_WRD;	!READ THE DATA WORD	
18304	:	8338			
18305	:	8339	incr NIB_PTR from 0 to 8 do	!LOOK AT 9 NIBBLES	
18306	:	8340			
18307	:	8341	if .PD_TEMP [.NIB_PTR] IS_NOT_SET	!FIND GOOD NIBBLES	
18308	:	8342	then		
18309	:	8343	begin		
18310	:	8344	XOR_LNG_WRD (.NIB_PTR, .TST_PAT, RESULT);	!XOR NIBBLE DATA WITH TST_PAT	
18311	:	8345			
18312	:	8346	if .RESULT<0, 4> neq ZERO	!SEE IF EQUAL	
18313	:	8347	then		
18314	:	8348	begin	!ERROR IF NEQ	
18315	:	8349	CMP THRESHOLD;	!COMPARE ERROR PRINT THRESHOLD	
18316	:	8350	ERRDF (86, SYNC, DUMPER);		

18318 ;ML4AD

18319 ;

18320

18321 ;

18322 ;

18323 ;

18324 ;

18325 ;

18326 ;

18327 ;

18328 ;

18329 ;

18330 ;

18331 ;

18332 ;

18333 ;

18334 ;

18335 ;

18336 ;

18337 ;

18338 ;

18339 ;

18340 ;

18341 ;

18342 ;

18343 ;

18344 ;

18348

18349

18353

18354

18355

18356

18357

18358

18359

18360

18361

18362

18363

18364

18365

18366

18367

18368

18369

18370

18371

18372

8351
8352
8353
8354
8355
8356
8357
8358
8359
8360
8361
8362
8363
8364
8365
8366
8367
8368
8369
8370
8371
8372
8373
8374

TEST CODE SECTION

```
PRINTB (FIV_FMT, WRD_24, WRD_25, WRD_10, WRD_12, FNC_5);
PRINTB (FMT_5, .TST_PAT, .RESULT, .NIB_PTR);
DODU_FLG = ONE;
end
end
end;
ENDSUB;
if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;
BG_PAT = not .BG_PAT;
TST_PAT = not .TST_PAT;
end;
ENDTST;
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (86)

```
PRINTB (FIV_FMT, WRD_24, WRD_25, WRD_10, WRD_12, FNC_5);
PRINTB (FMT_5, .TST_PAT, .RESULT, .NIB_PTR);
DODU_FLG = ONE;
end
```

!DROP THIS UNIT IF DODU IS_SET
!AND EXITS TEST

!COMPLIMENT BG_PAT
!COMPLIMENT TST_PAT AND REPEAT

ST34: .SBTTL ST34 TEST CODE SECTION

```
JSR R1,$$SAVE5
CMP -(SP),-(SP)
CLR P.CNT
CLR -(SP)
CLR R1
MOV #-1,R2
CLR -(SP)
1$: TRAP 2
BISB #40,@ML.REG+40
MOV ML.DUT,R5
BIC #177770,R5
BICB #7,@ML.REG+40
BISB R5,@ML.REG+40
MOV R1,@ML.REG+170
MOV R1,@ML.REG+200
MOV R1,@ML.REG+160
JSR PC,DAT.DM.XFER
MOV #61,@ML.REG
CLR R3
2$: MOV #1,R4
```

: DODU.FLG
: BG.PAT
: *,TST.PAT
: TWICE

: BG.PAT,*
: BG.PAT,*
: BG.PAT,*

: WD.CNT
: *,\$\$TMP2

8245

8298

8301

8302

8303

8305

8306

8307

8309

8310

8311

8312

8313

8315

8317

Address	OpCode	Op1	Op2	Op3	Op4	Instruction	Comments	Label
18430								
18431								
18432								
18433	065724	017767	127654	125444		MOV	2ML.REG+200,D2.TEMP	
18434	065732	017767	127626	125440		MOV	2ML.REG+160,E2.TEMP	
18435	065740	005004				CLR	R4	
18436	065742	010405				MOV	R4,R5	: NIB.PTR
18437	065744	006205				ASR	R5	: NIB.PTR,*
18438	065746	006205				ASR	R5	
18439	065750	006205				ASR	R5	
18440	065752	062705	015342			ADD	#PD.TEMP,R5	
18441	065756	010546				MOV	R5,-(SP)	
18442	065760	010446				MOV	R4,-(SP)	
18443	065762	042716	177770			BIC	#177770,(SP)	: NIB.PTR,*
18444	065766	012746	000001			MOV	#1,-(SP)	
18445	065772	005046				CLR	-(SP)	
18446	065774	004767	115634			JSR	PC,BLSGT2	
18447	066000	062706	000010			ADD	#10,SP	
18448	066004	005700				TST	R0	
18449	066006	001071				BNE	20\$	
18450	066010	010446				MOV	R4,-(SP)	: NIB.PTR,*
18451	066012	010246				MOV	R2,-(SP)	: TST.PAT,*
18452	066014	012746	000012			MOV	#12,-(SP)	
18453	066020	060616				ADD	SP,(SP)	: RESULT,*
18454	066022	004767	133044			JSR	PC,XOR.LNG.WRD	
18455	066026	032766	000017	000012		BIT	#17,12(SP)	: *,RESULT
18456	066034	001454				BEQ	19\$	
18457	066036	005267	127334			INC	P.CNT	
18458	066042	026767	127330	127330		CMP	P.CNT,LIMIT	
18459	066050	003403				BLE	18\$	
18460	066052	062706	000006			ADD	#6,SP	
18461	066056	000451				BR	21\$	
18462	066060	104455				TRAP	55	
18463	066062	000126				.WORD	126	
18464	066064	012750				.WORD	SYNC	
18465	066066	026302				.WORD	DUMPER	
18466	066070	012746	012216			MOV	#FNC.5,-(SP)	
18467	066074	012746	010650			MOV	#WRD.12,-(SP)	
18468	066100	012746	010630			MOV	#WRD.10,-(SP)	
18469	066104	012746	011002			MOV	#WRD.25,-(SP)	
18470	066110	012746	010774			MOV	#WRD.24,-(SP)	
18471	066114	012746	010304			MOV	#FIV.FMT,-(SP)	
18472	066120	012746	000006			MOV	#6,-(SP)	
18473	066124	010600				MOV	SP,R0	: SP,*
18474	066126	104414				TRAP	14	
18475	066130	010416				MOV	R4,(SP)	: NIB.PTR,*
18476	066132	016646	000030			MOV	30(SP),-(SP)	: RESULT,*
18477	066136	010246				MOV	R2,-(SP)	: TST.PAT,*
18478	066140	012746	006650			MOV	#FMT.5,-(SP)	
18479	066144	012746	000004			MOV	#4,-(SP)	
18480	066150	010600				MOV	SP,R0	: SP,*
18481	066152	104414				TRAP	14	
18482	066154	012766	000001	000036		MOV	#1,36(SP)	: *,DODU.FLG
18483	066162	062706	000026			ADD	#26,SP	
18484	066166	062706	000006			ADD	#6,SP	

18542 :ML4AD
18543 :
18544 :
18545 :

TEST CODE SECTION

8375 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (86)

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>M_4AD.BLI.4 '87)

18547 :ML4AD
18548 :
18549 :
18550 :
18551 :
18552 :
18553 :
18554 :
18555 :
18556 :
18557 :
18558 :
18559 :
18560 :
18561 :
18562 :
18563 :
18564 :
18565 :
18566 :
18567 :
18568 :
18569 :
18570 :
18571 :
18572 :
18573 :
18574 :
18575 :
18576 :
18577 :
18578 :
18579 :
18580 :
18581 :
18582 :
18583 :
18584 :
18585 :
18586 :
18587 :
18588 :
18589 :
18590 :
18591 :
18592 :
18593 :
18594 :
18595 :
18596 :
18597 :
18598 :
18599 :
18600 :
18601 :

8376
8377
8378
8379
8380
8381
8382
8383
8384
8385
8386
8387
8388
8389
8390
8391
8392
8393
8394
8395
8396
8397
8398
8399
8400
8401
8402
8403
8404
8405
8406
8407
8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427

TEST CODE SECTION

```
!
BGNTST:
!++
TEST NUMBER: TST 35
TEST NAME: SYNC DATA BUS CONTINUITY /READ PATH
TEST DESCRIPTION:
TEST THE CONTINUITY OF THE SYNCHRONOUS MODULE READ
DATA BUS BY:
1. VIA MBUS WRITE FUNCTION WRITE ONES INTO THE GOOD BLOCK.
2. VIA MBUS READ FUNCTION READ THE GOOD BLOCK FOR ONES.
3. REPEAT WITH COMPLIMENT DATA PATTERN.
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
TST_PAT, !TEST PATTERN
BG_PAT; !BACKGROUND PATTERN

CLR_THRESHOLD; !CLEAR ERROR PRINT THRESHOLD
DODD_FLG = ZERO;
TST_PAT = ONES;
BG_PAT = ZEROES;

incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
begin
CLR_MBUS;
BAI = ONE; !SET ON FIRST IO_BUF ADRS
IO_BUF = .TST_PAT; !FIRST IO_BUF ADRS GET TST_PAT
GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
MLCS1 = write; !DO A WRITE FUNCTION (WRITES THE TST_PAT)
TIME_OUT_LOOP;
BGNSDB;
CLR_MBUS;

incr IO_CNT from 0 to 255 do !LOAD IO_BUF WITH BG PAT
IO_BUF [.IO_CNT] = .BG_PAT;

GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
MLCS1 = read; !DO A READ FUNCTION (READ THE TST_PAT)
TIME_OUT_LOOP;
```


18603 :ML4AD

18604 :
18605 :
18606 : 8428
18607 : 8429
18608 : 8430
18609 : 8431
18610 : 8432
18611 : 8433
18612 : 8434
18613 : 8435
18614 : 8436
18615 : 8437
18616 : 8438
18617 : 8439
18618 : 8440
18619 : 8441
18620 : 8442
18621 : 8443
18622 : 8444
18623 : 8445
18624 : 8446
18625 : 8447
18626 : 8448
18627 : 8449
18628 : 8450
18629 : 8451
18630 : 8452
18631 : 8453
18635 :
18636 :

TEST CODE SECTION

incr IO_CNT from 0 to 255 do

if .IO_BUF [.IO_CNT] neq .TST_PAT
then

begin

 CMP THRESHOLD;

 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
then

begin

 DODU (.ML_LUN);

 DOCLN;

end;

TST_PAT = not .TST_PAT;

BG_PAT = not .BG_PAT;

end;

ENDTST;

29-Mar-1982 16:23:04

TOPS-20 Bliss-16 V2(212)

29-Mar-1982 16:21:03

PA:<NEALE>ML4AD.BLI.4 (87)

!READ THE IO_BUF FOR TEST PATTERN

!SEE IF IO_BUF WORD EQLS TST PAT

!ERROR AND SET DODU_FLG IF NEQ

!COMPARE ERROR PRINT THRESHOLD

!DROP THIS UNIT IF DODU_FLG IS SET

!COMPLIMENT TST_PAT

!COMPLIMENT BG_PAT AND REPEAT

18640	066302	004167	116304		\$T35:	.SBTTL	\$T35 TEST CODE SECTION	
18641	066306	005067	127064			JSR	R1,\$\$SAVES	
18642	066312	005046				CLR	P.CNT	
18643	066314	012701	177777			CLR	-(SP)	
18644	066320	005004				MOV	#-1,R1	
18645	066322	005005				CLR	R4	
18646	066324	152777	000040	127112	1\$:	CLR	R5	
18647	066332	016703	127474			BISB	#40,@ML.REG+40	
18648	066336	042703	177770			MOV	ML.DUT,R3	
18649	066342	142777	000007	127074		BIC	#177770,R3	
18650	066350	150377	127070			BICB	#7,@ML.REG+40	
18651	066354	152777	000010	127062		BISB	R3,@ML.REG+40	
18652	066362	010167	125434			BISB	#10,@ML.REG+40	
18653	066366	004767	131720			MOV	R1,IO.BUF	
18654	066372	012777	000061	127004		JSR	PC,GD.BLK.XFER	
18655	066400	105777	127050		2\$:	MOV	#61,@ML.REG	
18656	066404	100375				BPL	2\$	
18657	066406	104402			3\$:	TRAP	2	

8374
8403
8406
8407
8408
8410
8411

DODU.FLG
*.TST.PAT
BG.PAT
TWICE

TST.PAT,*

8413
8414
8415
8416

8417

Address	OpCode	Operand 1	Operand 2	Operand 3	Label	Instruction	Comments	Page
18659								
18660								
18661								
18662	066410	152777	000040	127026		BISB #40,@ML.RL,40		
18663	066416	016703	127410			MOV ML.DUT,R3		
18664	066422	042703	177770			BIC #177770,R3		
18665	066426	142777	000007	127010		BICB #7,@ML.REG+40		
18666	066434	150377	127004			BISB R3,@ML.REG+40		
18667	066440	005002				CLR R2		
18668	066442	010203			4\$:	MOV R2,R3	: IO.CNT	8421
18669	066444	006303				ASL R3	: IO.CNT,*	8422
18670	066446	010463	014022			MOV R4,IO.BUF(R3)	: BG.PAT,*	
18671	066452	005202				INC R2	: IO.CNT	
18672	066454	020227	000377			CMP R2,#377	: IO.CNT,*	8421
18673	066460	003770				BLE 4\$		
18674	066462	004767	131624			JSR PC,GD.BLK.XFER		
18675	066466	012777	000071	126710		MOV #71,@ML.REG		8424
18676	066474	105777	126754		5\$:	TSTB @ML.REG+50		8425
18677	066500	100375				BPL 5\$		
18678	066502	005002				CLR R2		
18679	066504	010203			6\$:	MOV R2,R3	: IO.CNT	8428
18680	066506	006303				ASL R3	: IO.CNT,*	8430
18681	066510	062703	014022			ADD #IO.BUF,R3		
18682	066514	021301				CMP (R3),R1	: *,TST.PAT	
18683	066516	001462				BEQ 7\$		
18684	066520	005267	126652			INC P.CNT		
18685	066524	026767	126646	126646		CMP P.CNT,LIMIT		8432
18686	066532	003060				BGT 8\$		
18687	066534	104455				TRAP 55		
18688	066536	000127				.WORD 127		8434
18689	066540	012750				.WORD SYNC		
18690	066542	026302				.WORD DUMPER		
18691	066544	012746	010730			MOV #WRD.19,-(SP)		
18692	066550	012746	012226			MOV #FNC.6,-(SP)		8435
18693	066554	012746	010766			MOV #WRD.23,-(SP)		
18694	066560	012746	010650			MOV #WRD.12,-(SP)		
18695	066564	012746	010630			MOV #WRD.10,-(SP)		
18696	066570	012746	011002			MOV #WRD.25,-(SP)		
18697	066574	012746	010774			MOV #WRD.24,-(SP)		
18698	066600	012746	010342			MOV #SEV.FMT,-(SP)		
18699	066604	012746	000010			MOV #10,-(SP)		
18700	066610	010600				MOV SP,R0	: SP,*	
18701	066612	104414				TRAP 14		
18702	066614	011316				MOV (R3),(SP)		
18703	066616	010146				MOV R1,-(SP)	: TST.PAT,*	8436
18704	066620	046616	000002			BIC 2(SP),(SP)		
18705	066624	040166	000002			BIC R1,2(SP)	: TST.PAT,*	
18706	066630	052616				BIS (SP)+,(SP)		
18707	066632	011346				MOV (R3)-,(SP)		
18708	066634	010146				MOV R1,-(SP)	: TST.PAT,*	
18709	066636	012746	006506			MOV #FMT.2,-(SP)		
18710	066642	012746	000004			MOV #4,-(SP)		
18711	066646	010600				MOV SP,R0	: SP,*	
18712	066650	104414				TRAP 14		
18713	066652	012766	000001	000032		MOV #1.32(SP)	: *,DODU.FLG	8437

18715
 18716
 18717
 18718 066660 062706 000032
 18719 066664 005202
 18720 066666 020227 000377
 18721 066672 003704
 18722 066674 104467
 18723 066676 006000
 18724 066700 103642
 18725 066702 021627 000001
 18726 066706 001004
 18727 066710 016700 127114
 18728 066714 104451
 18729 066716 104444
 18730 066720 005101
 18731 066722 005104
 18732 066724 005205
 18733 066726 020527 000001
 18734 066732 003002
 18735 066734 000167 177364
 18736 066740 005726
 18737 066742 000207

```

:ML4AD
:
TEST CODE SECTION
7$:  ADD #32,SP
    INC R2
    CMP R2,#377
    BLE 6$
8$:  TRAP 67
    ROR R0
    BLO 3$
    CMP (SP),#1
    BNE 9$
    MOV ML,LUN,R0
    TRAP 51
9$:  TRAP 44
    COM R1
    COM R4
    INC R5
    CMP R5,#1
    BGT 10$
    JMP 1$
10$: TST (SP)+
    RTS PC
  
```

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

8432
 8428
 8438
 8442
 8445
 8449
 8450
 8410
 8374

: Routine Size: 145 words
 : Maximum stack depth per invocation: 20 words

18738
 18739
 18740
 18745
 18746
 18750
 18751
 18755 066744
 18756 066744 004767 177332
 18757 066750 104466
 18758 066752 006000
 18759 066754 103773
 18760 066756 000207

```

.SBTTL T35 TEST CODE SECTION
T35::
1$:  JSR PC,$T35
    TRAP 66
    ROR R0
    BLO 1$
    RTS PC
  
```

8451

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

18761
 18762
 18763
 18768
 18769 : 8454 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (88)

18771 : ML4AD
18772 :
18773 :
18774 : 8455
18775 : 8456
18776 : 8457
18777 : 8458
18778 : 8459
18779 : 8460
18780 : 8461
18781 : 8462
18782 : 8463
18783 : 8464
18784 : 8465
18785 : 8466
18786 : 8467
18787 : 8468
18788 : 8469
18789 : 8470
18790 : 8471
18791 : 8472
18792 : 8473
18793 : 8474
18794 : 8475
18795 : 8476
18796 : 8477
18797 : 8478
18798 : 8479
18799 : 8480
18800 : 8481
18801 : 8482
18802 : 8483
18803 : 8484
18804 : 8485
18805 : 8486
18806 : 8487
18807 : 8488
18808 : 8489
18809 : 8490
18810 : 8491
18811 : 8492
18812 : 8493
18813 : 8494
18814 : 8495
18815 : 8496
18816 : 8497
18817 : 8498
18818 : 8499
18819 : 8500
18820 : 8501
18821 : 8502
18822 : 8503
18823 : 8504
18824 : 8505
18825 : 8506

TEST CODE SECTION

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

```

18827 :ML4AD
18828 :
18829 :
18830 :      8507      WRD_CNT,
18831 :      8508      NIB_IGNORE : bitvector [16],
18832 :      8509      ERR_FLG,
18833 :      8510      BAD_NIB_CNT,
18834 :      8511      PASS_CNT,
18835 :      8512      NIB_PAT;
18836 :      8513
18837 :      8514 CLR_THRESHOLD;
18838 :      8515 DODD_FLG = ZERO;
18839 :      8516 WRD_CNT = ZERO;
18840 :      8517
18841 :      8518 incr CNT from 0 to 63 do
18842 :      8519
18843 :      8520      incr PAT_CNT from -1 to 11 by 4 do
18844 :      8521      begin
18845 :      8522      (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1;
18846 :      8523      (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2;
18847 :      8524      (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3;
18848 :      8525      (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4;
18849 :      8526      WRD_CNT = .WRD_CNT + 1;
18850 :      8527      end;
18851 :      8528
18852 :      8529 BGNSUB;
18853 :      8530 CLR_MBUS;
18854 :      8531 GD_BLK_XFER ();
18855 :      8532 MLCs1 = write;
18856 :      8533 TIME_OUT_LOOP;
18857 :      8534 CLR_MBUS;
18858 :      8535 NIB_IGNORE = ZEROES;
18859 :      8536 PASS_CNT = ZEROES;
18860 :      8537 NIB_PAT = ZEROES;
18861 :      8538 BAD_NIB_CNT = ZEROES;
18862 :      8539 DAT_DM_XFER ();
18863 :      8540 MLCs1 = read;
18864 :      8541 DELAY (ONE_US);
18865 :      8542
18866 :      8543 do
18867 :      8544      begin
18868 :      8545      PD_TEMP = .MLPD;
18869 :      8546      DAT_CLK = ONE;
18870 :      8547      DELAY (ONE_US);
18871 :      8548      RD_LNG_WRD;
18872 :      8549
18873 :      8550      incr NIB_PTR from 0 to 8 do
18874 :      8551      begin
18875 :      8552
18876 :      8553      if .PD_TEMP [.NIB_PTR] IS_NOT_SET
18877 :      8554      then
18878 :      8555      begin
18879 :      8556
18880 :      8557      if .NIB_IGNORE [.NIB_PTR] IS_NOT_SET
18881 :      8558      then

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (88)

```

!WORD COUNT
!NIBBLE IGNORE FLAGS
!ERROR FLAG
!BAD NIBBLE COUNT
!PASS COUNT
!NIBBLE PATTERN

!CLEAR ERROR PRINT THRESHOLD

!LOAD 64 WORDS WITH REPEATING COUNTS OF 0-63

!LOAD NIBBLES IN WORD WITH REPEATING COUNTS OF 0-63

!LOAD FIRST NIBBLE IN WORD
!LOAD SECOND NIBBLE IN WORD
!LOAD THIRD NIBBLE IN WORD
!LOAD FORTH NIBBLE IN WORD
!INCREMENT TO NEXT WORD

!SET UP A GOOD BLOCK XFERR
!DO A WRITE FUNCTION

!SET UP A DATA DIAG MODE AT THE GOOD BLOCK
!DO A READ FUNCTION

!LOOP UNTIL THE BLOCK IS READ OR 9 BAD NIBBLES FOUND

!GET THE PROM DATA
!CLOCK OUT THE DATA WORD

!READ THE DATA DIAG REGISTERS

!LOOK AT 9 NIBBLES

!FIND GOOD NIBBLES

!SEE IF THIS NIBBLE FOUND BAD BEFORE

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (88)

```

18883 :ML4AD
18884 :
18885 :
18886 :      8559
18887 :      8560
18888 :      8561
18889 :      8562
18890 :      8563
18891 :      8564
18892 :      8565
18893 :      8566
18894 :      8567
18895 :      8568
18896 :      8569
18897 :      8570
18898 :      8571
18899 :      8572
18900 :      8573
18901 :      8574
18902 :      8575
18903 :      8576
18904 :      8577
18905 :      8578
18906 :      8579
18907 :      8580
18908 :      8581
18909 :      8582
18910 :      8583
18911 :      8584
18912 :      8585
18913 :      8586
18914 :      8587
18915 :      8588
18916 :      8589
18917 :      8590
18918 :      8591
18919 :      8592
18920 :      8593
18921 :      8594
18922 :      8595
18923 :      8596
18927 :
18928 :
18932 066760 004167 115626
18933 066764 162706 000006
18934 066770 005067 126402
18935 066774 005046
18936 066776 005003
18937 067000 005002

TEST CODE SECTION

begin
TST_LNG_WRD (.NIB_PTR, .NIB_PAT, ERR_FLG);      !TEST THE NIBBLE IF NEVER FOUND BAD
if .ERR_FLG IS_SET
then
begin
CMP THRESHOLD;      !ERROR AND SET DODU_FLG IS SET
ERRDF (88, ARR DAT, DUMPER);      !COMPARE ERROR PRINT THRESHOLD
PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_5, WRD_19);
DODU_FLG = ONE;
end
end

else
begin
NIB_IGNORE [.NIB_PTR] = ONE;      !THIS NIBBLE IS BAD
BAD_NIB_CNT = .BAD_NIB_CNT + 1;      !SET THIS NIBBLE NIB IGNORE FLAG
end;      !INCREMENT BAD NIB COUNT

NIB_PAT = .NIB_PAT + 1;      !INCREMENT NIB PAT
end;

PASS_CNT = .PASS_CNT + 1;      !INCREMENT PASS COUNT
end
until (.PASS_CNT eql 113) or (.BAD_NIB_CNT eql 9);      !REPEAT UNTIL COMPLETE
ENDSUB;

if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);      !DROP THIS UNIT IF DODU_FLG SET
DOCLN;
end;

ENDTST;

.SBTTL $T36 TEST CODE SECTION
$T36: JSR R1,$SAVE5
SUB #6,SP
CLR P,CNT
CLR -(SP)
CLR R3
CLR R2
:
: DODU.FLG
: WRD.CNT
: CNT

```

8453
8512
8515
8516
8518

Address	Op Code	Op 1	Op 2	Op 3	Instruction	Comments	Line No.
18939							
18940							
18941							
18942	067002	012700	177777		MOV #1,R0	: *,PAT.CNT	
18943	067006	010301			MOV R3,R1	: WRD.CNT,*	8520
18944	067010	006301			ASL R1		8522
18945	067012	062701	014022		ADD #10,BUF,R1		
18946	067016	010005			MOV R0,R5	: PAT.CNT,*	
18947	067020	005205			INC R5		
18948	067022	042705	177760		BIC #177760,R5		
18949	067026	142711	000017		BICB #17,(R1)		
18950	067032	150511			BISB R5,(R1)		
18951	067034	010005			MOV R0,R5	: PAT.CNT,*	
18952	067036	062705	000002		ADD #2,R5		8523
18953	067042	006305			ASL R5		
18954	067044	006305			ASL R5		
18955	067046	006305			ASL R5		
18956	067050	006305			ASL R5		
18957	067052	042705	177417		BIC #177417,R5		
18958	067056	142711	000360		BICB #360,(R1)		
18959	067062	150511			BISB R5,(R1)		
18960	067064	010005			MOV R0,R5	: PAT.CNT,*	
18961	067066	062705	000003		ADD #3,R5		8524
18962	067072	000305			SWAB R5		
18963	067074	042705	170377		BIC #170377,R5		
18964	067100	042711	007400		BIC #7400,(R1)		
18965	067104	050511			BIS R5,(R1)		
18966	067106	010005			MOV R0,R5	: PAT.CNT,*	
18967	067110	062705	000004		ADD #4,R5		8525
18968	067114	000305			SWAB R5		
18969	067116	006305			ASL R5		
18970	067120	006305			ASL R5		
18971	067122	006305			ASL R5		
18972	067124	006305			ASL R5		
18973	067126	042705	007777		BIC #7777,R5		
18974	067132	042711	170000		BIC #170000,(R1)		
18975	067136	050511			BIS R5,(R1)		
18976	067140	005203			INC R3	: WRD.CNT	8526
18977	067142	062700	000004		ADD #4,R0	: *,PAT.CNT	8520
18978	067146	020027	000013		CMP R0,#13	: PAT.CNT,*	
18979	067152	003715			BLE 2\$		
18980	067154	005202			INC R2	: CNT	8518
18981	067156	020227	000077		CMP R2,#77	: CNT,*	
18982	067162	003707			BLE 1\$		
18983	067164	104402			TRAP 2		
18984	067166	152777	000040	126250	BISB #40,@ML.REG+40		8527
18985	067174	016702	126632		MOV ML,DUT,R2		8529
18986	067200	042702	177770		BIC #177770,R2		
18987	067204	142777	000007	126232	BICB #7,@ML.REG+40		
18988	067212	150277	126226		BISB R2,@ML.REG+40		
18989	067216	004767	131070		JSR PC,GD,BLK.XFER		
18990	067222	012777	000061	126154	MOV #61,@ML.REG		8531
18991	067230	105777	126220		TSTB @ML.REG+50		8532
18992	067234	100375			BPL 4\$		
18993	067236	152777	000040	126200	BISB #40,@ML.REG+40		8533

```

18995                                :ML4AD
18996                                :
18997                                :
18998 067244 016702 126562          MOV     ML.DUT,R2
18999 067250 042702 177770          BIC     #177770,R2
19000 067254 142777 000007 126162  BICB   #7,@ML.REG+40
19001 067262 150277 126156          BISB   R2,@ML.REG+40
19002 067266 005066 000004          CLR     4(SP)
19003 067272 005004                   : NIB.IGNORE                      8535
19004 067274 005003                   : PASS.CNT                          8536
19005 067276 005005                   : NIB.PAT                            8537
19006 067300 004767 127106          CLR     R4
19007 067304 012777 000071 126072  CLR     R3
19008 067312 012701 000001          CLR     R5
19009 067316 001411                    : BAD.NIB.CNT                       8538
19010 067320 016702 112572          JSR    PC,DAT.DM.XFER
19011 067324 001404                    :                                     8539
19012 067326 005066 000006          MOV     #71,@ML.REG
19013 067332 005302                    :                                     8540
19014 067334 001374                    : *,$$TMP2                           8541
19015 067336 005301                    :
19016 067340 000766                    :                                     :
19017 067342 017767 126266 125772  MOV     @ML.REG+230,PD.TEMP
19018 067350 152777 000020 126146  BISB   #20,@ML.REG+120
19019 067356 012701 000001          :                                     :
19020 067362 001411                    : *,$$TMP2                           8545
19021 067364 016702 112526          :                                     :
19022 067370 001404                    :                                     8546
19023 067372 005066 000006          : *,$$TMP1                           8547
19024 067376 005302                    :
19025 067400 001374                    : $$TMP
19026 067402 005301                    : $$TMP1
19027 067404 000766                    :
19028 067406 017767 126162 123760  BR      5$
19029 067414 017767 126164 123754  MOV     @ML.REG+170,D1.TEMP
19030 067422 017767 126136 123750  MOV     @ML.REG+200,D2.TEMP
19031 067430 005001                    :
19032 067432 010100                    : *,$$TMP2
19033 067434 006200                    :                                     :
19034 067436 006200                    : *,$$TMP1
19035 067440 006200                    :
19036 067442 012702 000004          : $$TMP
19037 067446 060602                    : $$TMP1
19038 067450 060002                    :
19039 067452 010046                    :
19040 067454 062716 015342          BR      9$
19041 067460 010146                    :
19042 067462 042716 177770          MOV     @ML.REG+160,E2.TEMP
19043 067466 012746 000001          :
19044 067472 005046                    : NIB.PTR                          8550
19045 067474 004767 114134          : NIB.PTR,*                         8553
19046 067500 062706 000910          :
19047 067504 005700                    :
19048 067506 001077                    :
19049 067510 010246                    : NIB.IGNORE,*                      8557

```

8557

Address	OpCode	Operand1	Operand2	Operand3	Label	Comment	Page
19051					:ML4AD		
19052					:	TEST CODE SECTION	
19053							
19054	067512	010146					
19055	067514	042716	177770			: NIB.PTR,*	
19056	067520	012746	000001				
19057	067524	005046					
19058	067526	004767	114102				
19059	067532	062706	000010				
19060	067536	005700					
19061	067540	001076					
19062	067542	010146					
19063	067544	010346				: NIB.PTR,*	8560
19064	067546	012746	000010			: NIB.PAT,*	
19065	067552	060616					
19066	067554	004767	130616			: ERR.FLG,*	
19067	067560	026627	000010	000001			
19068	067566	001044				: ERR.FLG,*	8562
19069	067570	005267	125602				
19070	067574	026767	125576	125576			8564
19071	067602	003403					
19072	067604	062706	000006				
19073	067610	000457					
19074	067612	104455			14\$:		
19075	067614	000130					8566
19076	067616	013012					
19077	067620	026302					
19078	067622	012746	010730				
19079	067626	012746	012216				8567
19080	067632	012746	010650				
19081	067636	012746	010630				
19082	067642	012746	011270				
19083	067646	012746	012414				
19084	067652	012746	010322				
19085	067656	012746	000007				
19086	067662	010600					
19087	067664	104414				: SP,*	
19088	067666	012766	000001	000026			
19089	067674	062706	000020			: *,DODU.FLG	8568
19090	067700	062706	000006		15\$:		8564
19091	067704	000414					8559
19092	067706	010246			16\$:		8553
19093	067710	010146					8576
19094	067712	042716	177770				
19095	067716	012746	000001			: NIB.PTR,*	
19096	067722	011646					
19097	067724	004767	114142				
19098	067730	005205					
19099	067732	062706	000010			: BAD.NIB.CNT	8577
19100	067736	005203			17\$:		8575
19101	067740	005201				: NIB.PAT	8580
19102	067742	020127	000010			: NIB.PTR	8550
19103	067746	003631				: NIB.PTR,*	
19104	067750	005204			18\$:		
19105	067752	020427	000161			: PASS.CNT	8583
						: PASS.CNT,*	8585

19107
 19108
 19109
 19110 067756 001405
 19111 067760 020527 000011
 19112 067764 001402
 19113 067766 000167 177350
 19114 067772 104467
 19115 067774 006000
 19116 067776 103002
 19117 070000 000167 177160
 19118 070004 021627 000001
 19119 070010 001004
 19120 070012 016700 126012
 19121 070016 104451
 19122 070020 104444
 19123 070022 062706 000010
 19124 070026 000207

```

:ML4AD
:
TEST CODE SECTION
:
19$: BEQ 19$
      CMP R5,#11
      BEQ 19$
      JMP 8$
      TRAP 67
      ROR R0
      BHIS 20$
      JMP 3$
20$: CMP (SP),#1
      BNE 21$
      MOV ML,LUN,R0
      TRAP 51
      TRAP 44
21$: ADD #10,SP
      RTS PC
  
```

: BAD.NIB.CNT,*

: DODU.FLG,*

8589

8592

8453

```

: Routine Size: 276 words
: Maximum stack depth per invocation: 21 words
  
```

19125
 19126
 19127
 19132
 19133
 19137
 19138
 19142 070030
 19143 070030 004767 176724
 19144 070034 104466
 19145 070036 006000
 19146 070040 103773
 19147 070042 000207
 19148
 19149
 19150
 19155
 19156
 19157 :

T36:: .SBTTL T36 TEST CODE SECTION

```

1$: JSR PC,$T36
    TRAP 66
    ROR R0
    BLO 1$
    RTS PC
  
```

8594

```

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

8597 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (89)

19159 :ML4AD
19160 :
19161 :
19162 :
19163 :
19164 :
19165 :
19166 :
19167 :
19168 :
19169 :
19170 :
19171 :
19172 :
19173 :
19174 :
19175 :
19176 :
19177 :
19178 :
19179 :
19180 :
19181 :
19182 :
19183 :
19184 :
19185 :
19186 :
19187 :
19188 :
19189 :
19190 :
19191 :
19192 :
19193 :
19194 :
19195 :
19196 :
19197 :
19198 :
19199 :
19200 :
19201 :
19202 :
19203 :
19204 :
19205 :
19206 :
19207 :
19208 :
19209 :
19210 :
19211 :
19212 :
19213 :

8598
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619
8620
8621
8622
8623
8624
8625
8626
8627
8628
8629
8630
8631
8632
8633
8634
8635
8636
8637
8638
8639
8640
8641
8642
8643
8644
8645
8646
8647
8648
8649

TEST CODE SECTION

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_THRESHOLD;
CLR_MBUS;
DODU_FLG = ZERO;
WRD_CNT = ZERO;

!DROP UNIT FLAG
!WORD COUNT
!PATTERN INCREMENT
!SIZE EXPRESSION
!POSITIONAL EXPRESSION
!TEMPORARY STORAGE LOCATION
!CLEAR ERROR PRINT THRESHOLD

```

19215 :ML4AD
19216 :
19217 :
19218 :      8650  incr COUNT from 0 to 63 do
19219 :      8651
19220 :      8652      incr PAT_CNT from -1 to 11 by 4 do
19221 :      8653      begin
19222 :      8654          (IO_BUF [.WRD_CNT])<0, 4> = .PAT_CNT + 1;      !LOAD FIRST NIBBLE IN WORD
19223 :      8655          (IO_BUF [.WRD_CNT])<4, 4> = .PAT_CNT + 2;      !LOAD SECOND NIBBLE IN WORD
19224 :      8656          (IO_BUF [.WRD_CNT])<8, 4> = .PAT_CNT + 3;      !LOAD THIRD NIBBLE IN WORD
19225 :      8657          (IO_BUF [.WRD_CNT])<12, 4> = .PAT_CNT + 4;      !LOAD FORTH NIBBLE IN WORD
19226 :      8658          WRD_CNT = .WRD_CNT + 1;      !INCREMENT TO NEXT WORD
19227 :      8659      end;
19228 :      8660
19229 :      8661  GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
19230 :      8662  MLCS1 = write;      !DO A WRITE FUNCTION
19231 :      8663  TIME_OUT_LOOP;
19232 :      8664  BGNSUB;
19233 :      8665
19234 :      8666  incr IO_CNT from 0 to 255 do      !CLEAR OUT IO_BUF
19235 :      8667      IO_BUF [.IO_CNT] = ZEROES;
19236 :      8668
19237 :      8669  CLR_MBUS;
19238 :      8670  GD_BLK_XFER ();      !SET UP A GOOD BLOCK XFERR
19239 :      8671  MLCS1 = read;      !DO A READ FUNCTION
19240 :      8672  TIME_OUT_LOOP;
19241 :      8673  CLR_MBUS;
19242 :      8674  SIZ_EXP = 4;      !FIELD SIZE FOR NIBBLES ALWAYS 4 BITS
19243 :      8675  WRD_CNT = 0;
19244 :      8676
19245 :      8677  incr COUNT from 0 to 63 do      !READ 64 WORDS IN IO_BUF
19246 :      8678
19247 :      8679      incr PAT_CNT from -1 to 11 by 4 do      !READ REPEATING COUNTS OF 0-63
19248 :      8680      begin
19249 :      8681          POS_EXP = ZERO;      !FIELD SELECTOR SELECTS THE FOUR NIBBLES
19250 :      8682          PAT_INC = ONE;
19251 :      8683          TEMP = .IO_BUF [.WRD_CNT];      !GET A WORD OUT OF IO_BUF
19252 :      8684
19253 :      8685          incr CNT from 0 to 3 do      !READ THE FOUR NIBBLES IN WORD
19254 :      8686          begin
19255 :      8687
19256 :      8688          if .TEMP<.POS_EXP, .SIZ_EXP> neq (.PAT_CNT + .PAT_INC)      !COMPARE NIBBLE WITH RESPECTIVE 0-63 CNT
19257 :      8689          then
19258 :      8690          then
19259 :      8691          begin      !ERROR AND SET DODU_FLG IF NEQ
19260 :      8692          CMP_THRESHOLD;      !COMPARE ERROR PRINT THRESHOLD
19261 :      8693          ERRDF (89, ARR_DAT, DUMPER);
19262 :      8694          PRINTB (SIX_FMT, FNC_18, WRD_50, WRD_10, WRD_12, FNC_6, WRD_19);
19263 :      8695          DODU_FLG = ONE;
19264 :      8696          end;
19265 :      8697
19266 :      8698          POS_EXP = .POS_EXP + 4;      !POINT TO THE NEXT NIBBLE IN WORD
19267 :      8699          PAT_INC = .PAT_INC + 1;      !INCREMENT THE 0-63 COUNT
19268 :      8700      end;
19269 :      8701

```

```

19271 :ML4AD
19272 :
19273 : TEST CODE SECTION
19274 :      8702      WRD_CNT = .WRD_CNT + 1;
19275 :      8703      end;
19276 :      8704
19277 :      8705      ENDSUB;
19278 :      8706
19279 :      8707      if .DODU_FLG IS_SET
19280 :      8708      then
19281 :      8709          begin
19282 :      8710              DODU (.ML_LUN);
19283 :      8711              DOCLN;
19284 :      8712              end;
19285 :      8713
19286 :      8714      ENDTST;
19290 :
19291 :

```

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (89)
!GET THE NEXT IO_BUF WORD
!DROP THIS UNIT IF DODU_FLG SET

```

19295	070044	004167	114542		.SBTTL	\$T37 TEST CODE SECTION		
19296	070050	162706	000012		\$T37:	JSR	R1,\$SAVE5	
19297	070054	005067	125316			SUB	#12,SP	:
19298	070060	152777	000040	125356		CLR	P.CNT	:
19299	070066	016705	125740			BISB	#40,@ML.REG+40	:
19300	070072	042705	177770			MOV	ML.DUT,R5	:
19301	070076	142777	000007	125340		BIC	#177770,R5	
19302	070104	150577	125334			BICB	#7,@ML.REG+40	
19303	070110	005066	000010			BISB	R5,@ML.REG+40	
19304	070114	005066	000002			CLR	10(SP)	: DODU.FLG
19305	070120	005002				CLR	2(SP)	: WRD.CNT
19306	070122	012703	177777		1\$:	CLR	R2	: COUNT
19307	070126	016604	000002		2\$:	MOV	#-1,R3	: *,PAT.CNT
19308	070132	006304				MOV	2(SP),R4	: WRD.CNT,*
19309	070134	062704	014022			ASL	R4	
19310	070140	010305				ADD	#10.BUF,R4	
19311	070142	005205				MOV	R3,R5	: PAT.CNT,*
19312	070144	042705	177760			INC	R5	
19313	070150	142714	000017			BIC	#177760,R5	
19314	070154	150514				BICB	#17,(R4)	
19315	070156	010305				BISB	R5,(R4)	
19316	070160	062705	000002			MOV	R3,R5	: PAT.CNT,*
19317	070164	006305				ADD	#2,R5	
19318	070166	006305				ASL	R5	
19319	070170	006305				ASL	R5	
19320	070172	006305				ASL	R5	
19321	070174	042705	177417			ASL	R5	
19322	070200	142714	000360			BIC	#177417,R5	
19323	070204	150514				BICB	#360,(R4)	
19324	070206	010305				BISB	R5,(R4)	
19325	070210	062705	000003			MOV	R3,R5	: PAT.CNT,*
						ADD	#3,R5	8656

Address	OpCode	Operand1	Operand2	OpCode	Operand1	Operand2	Comments	Count
19327				SWAB	R5			
19328				BIC	#170377,R5			
19329				BIC	#7400,(R4)			
19330	070214	000305		BIS	R5,(R4)			
19331	070216	042705	170377	MOV	R3,R5			
19332	070222	042714	007400	ADD	#4,R5		: PAT.CNT,*	8657
19333	070226	050514		SWAB	R5			
19334	070230	010305		ASL	R5			
19335	070232	062705	000004	ASL	R5			
19336	070236	000305		ASL	R5			
19337	070240	006305		ASL	R5			
19338	070242	006305		BIC	#7777,R5			
19339	070244	006305		BIC	#170000,(R4)			
19340	070246	006305		BIS	R5,(R4)			
19341	070250	042705	007777	INC	2(SP)		: WRD.CNT	8658
19342	070254	042714	170000	ADD	#4,R3		: *,PAT.CNT	8652
19343	070260	050514		CMP	R3,#13		: PAT.CNT,*	
19344	070262	005266	000002	BLE	2\$			
19345	070266	062703	000004	INC	R2		: COUNT	8650
19346	070272	020327	000013	CMP	R2,#77		: COUNT,*	
19347	070276	003713		BLE	1\$			
19348	070300	005202		JSR	PC,GD.BLK.XFER			
19349	070302	020227	000077	MOV	#61,@ML.REG			8661
19350	070306	003705		TSTB	@ML.REG+50			8662
19351	070310	004767	127776	BPL	3\$			
19352	070314	012777	000061	TRAP	2			
19353	070322	105777	125126	3\$: CLR	R3		: IO.CNT	8663
19354	070326	100375		5\$: MOV	R3,R4		: IO.CNT,*	8666
19355	070330	104402		ASL	R4			8667
19356	070332	005003		CLR	IO.BUF(R4)			
19357	070334	010304		5\$: INC	R3		: IO.CNT	8666
19358	070336	006304		CMP	R3,#377		: IO.CNT,*	
19359	070340	005064	014022	BLE	5\$			
19360	070344	005203		BISB	#40,@ML.REG+40			
19361	070346	020327	000377	MOV	ML.DUT,R4			8667
19362	070352	003770		BIC	#177770,R4			
19363	070354	152777	000040	125062	BICB	#7,@ML.REG+40		
19364	070362	016704	125444	125044	BISB	R4,@ML.REG+40		
19365	070366	042704	177770	125044	JSR	PC,GD.BLK.XFER		
19366	070372	142777	000007	125044	MOV	#71,@ML.REG		
19367	070400	150477	125040	124766	6\$: TSTB	@ML.REG+50		8670
19368	070404	004767	127702	124766	6\$: BPL	6\$		8671
19369	070410	012777	000071	124766	6\$: BISB	#40,@ML.REG+40		
19370	070416	105777	125032	125012	6\$: MOV	ML.DUT,R4		8672
19371	070422	100375		125012	6\$: BIC	#177770,R4		
19372	070424	152777	000040	125012	6\$: BICB	#7,@ML.REG+40		
19373	070432	016704	125374	124774	6\$: BISB	R4,@ML.REG+40		
19374	070436	042704	177770	124774	6\$: MOV	#4,6(SP)		
19375	070442	142777	000007	124774	6\$: CLR	2(SP)	: *,SIZ.EXP	8674
19376	070450	150477	124770	000006	6\$: CLR	R1	: WRD.CNT	8675
19377	070454	012766	000004	000006	6\$: CLR	R1	: COUNT	8677
19378	070462	005066	000002	000006	6\$: MOV	#-1,R3	: *,PAT.CNT	8679
19379	070466	005001		000006	6\$: CLR	R5	: POS.EXP	8681
19380	070470	012703	177777	7\$:				
19381	070474	005005		8\$:				

19439
19440
19441
19442 070744 10445
19443 070746 104444
19444 070750 062706 000012
19445 070754 000207
19446
19447
19448
19453
19454
19458
19459
19463 070756
19464 070756 004767 177062
19465 070762 104466
19466 070764 006000
19467 070766 103773
19468 070770 000207
19469
19470
19471
19476
19477
19478 ;

:ML4AD
:
TEST CODE SECTION

12\$: TRAP 51
TRAP 44
ADD #12.SP
RTS PC

: Routine Size: 229 words
: Maximum stack depth per invocation: 19 words

.SBTTL T37 TEST CODE SECTION

T37::
1\$: JSR PC,\$T37
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

8596

8712

8715 !<BLF/PAGE>

19480 :ML4AD
19481 :
19482 :
19483 :
19484 :
19485 :
19486 :
19487 :
19488 :
19489 :
19490 :
19491 :
19492 :
19493 :
19494 :
19495 :
19496 :
19497 :
19498 :
19499 :
19500 :
19501 :
19502 :
19503 :
19504 :
19505 :
19506 :
19507 :
19508 :
19509 :
19510 :
19511 :
19512 :
19513 :
19514 :
19515 :
19516 :
19517 :
19518 :
19519 :
19520 :
19521 :
19522 :
19523 :
19524 :
19525 :
19526 :
19527 :
19528 :
19529 :
19530 :
19531 :
19532 :
19533 :
19534 :

8716
8717
8718
8719
8720
8721
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

TEST CODE SECTION

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.

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (90)

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (90)

```

19536 :ML4AD
19537 :
19538 :
19539 :      8768  !--
19540 :      8769
19541 :      8770  Local
19542 :      8771      SAV NIB,
19543 :      8772      DODD FLG,
19544 :      8773      NIB_BIT,
19545 :      8774      TST_PAT,
19546 :      8775      ALL_ONES_1,
19547 :      8776      ALL_ONES_2,
19548 :      8777      STK_PTR,
19549 :      8778      COUNT;
19550 :      8779
19551 :      8780  CLR_THRESHOLD;
19552 :      8781  BGNSUB;
19553 :      8782  CLR_MBUS;
19554 :      8783  DODD_FLG = ZERO;
19555 :      8784  TST_PAT = ONE;
19556 :      8785
19557 :      8786  incr CNT from 0 to 15 do
19558 :      8787      begin
19559 :      8788      IO_BUF [.CNT] = not .TST_PAT;
19560 :      8789      TST_PAT = .TST_PAT^ONE;
19561 :      8790      end;
19562 :      8791
19563 :      8792  GD_BLK_XFER ();
19564 :      8793  MLCS1 = write;
19565 :      8794  TIME_OUT_LOOP;
19566 :      8795  STRIPPER(21, 8);
19567 :      8796  STK_PTR = -1;
19568 :      8797  NIB_BIT = ONE;
19569 :      8798  ALL_ONES_1 = ZERO;
19570 :      8799  ALL_ONES_2 = 3;
19571 :      8800
19572 :      8801  incr BY_FOUR_WRDS from 0 to 3 do
19573 :      8802      begin
19574 :      8803
19575 :      8804      incr BY_ONE_WRD from 0 to 3 do
19576 :      8805      begin
19577 :      8806      COUNT = ZERO;
19578 :      8807
19579 :      8808      until .COUNT eql .ALL_ONES_1 do
19580 :      8809      begin
19581 :      8810      COUNT = .COUNT + 1;
19582 :      8811      STK_PTR = .STK_PTR + 1;
19583 :      8812
19584 :      8813      if (.stack [.STK_PTR]) neq %o'000017'
19585 :      8814      then
19586 :      8815      begin
19587 :      8816      CMP_THRESHOLD;
19588 :      8817      ERRDF (90, SYNC, DUMPER);
19589 :      8818      PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
19590 :      8819      PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STK_PTR);
  
```

```

!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

!CLEAR ERROR PRINT THRESHOLD

!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
!COMPARE ERROR PRINT THRESHOLD
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (90)

```

19592 :ML4AD
19593 :
19594 :
19595 :      8820          DODU_FLG = ONE;
19596 :      8821          end;
19597 :      8822
19598 :      8823          end;
19599 :      8824
19600 :      8825          STK_PTR = .STK_PTR + 1;          !INCREMENT THE STACK POINTER
19601 :      8826          SAV_NIB = ( not .NIB_BIT) and (%0'000017');          !GENERATE THE SHIFTED BIT
19602 :      8827
19603 :      8828          if (.stack [.STK_PTR]) neq (.SAV_NIB)          !COMPARE STACK TO SHIFTED BIT
19604 :      8829          then
19605 :      8830          begin
19606 :      8831          CMP THRESHOLD;          !ERROR AND SET DODU_FLG IF NEQ
19607 :      8832          ERRDF (91, SYNC, DUMPER);          !COMPARE ERROR PRINT THRESHOLD
19608 :      8833          PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
19609 :      8834          PRINTB (FMT_15, .STK_PTR);
19610 :      8835          PRINTB (FMT_5, .SAV_NIB, .stack [.STK_PTR]);
19611 :      8836          DODU_FLG = ONE;
19612 :      8837          end;
19613 :      8838
19614 :      8839          COUNT = ZEROES;          !CLEAR COUNT
19615 :      8840
19616 :      8841          until .COUNT eql .ALL_ONES_2 do          !READ X NUMBER OF '17' NIBBLE PAT
19617 :      8842          begin
19618 :      8843          COUNT = .COUNT + 1;          !INCREMENT COUNT
19619 :      8844          STK_PTR = .STK_PTR + 1;          !INCREMENT STACK POINTER
19620 :      8845
19621 :      8846          if (.stack [.STK_PTR]) neq %0'000017'          !COMPARE STACK POINTER WITH '17'
19622 :      8847          then
19623 :      8848          begin
19624 :      8849          CMP THRESHOLD;          !ERROR AND SET DODU_FLG IF SET
19625 :      8850          ERRDF (92, SYNC, DUMPER);          !COMPARE ERROR PRINT THRESHOLD
19626 :      8851          PRINTB (SIX_FMT, WRD_23, WRD_39, PHR_4, WRD_12, FNC_5, WRD_19);
19627 :      8852          PRINTB (FMT_5, ONES, .stack [.STK_PTR], .STR_PTR);
19628 :      8853          DODU_FLG = ONE;
19629 :      8854          end;
19630 :      8855
19631 :      8856          end;
19632 :      8857
19633 :      8858          NIB_BIT = .NIB_BIT^ONE;          !SHIFT THE SHIFTED NIBBLE BIT
19634 :      8859          end;
19635 :      8860
19636 :      8861          NIB_BIT = ONE;          !RESET THE SHIFTED NIBBLE BIT
19637 :      8862          ALL_ONES_1 = .ALL_ONES_1 + 1;          !READ ONE MORE '17' PATTERN
19638 :      8863          ALL_ONES_2 = .ALL_ONES_2 - 1;          !READ ONE LESS '17' PATTERN
19639 :      8864          end;
19640 :      8865
19641 :      8866          ENDSUB;
19642 :      8867
19643 :      8868          if .DODU_FLG IS_SET
19644 :      8869          then
19645 :      8870          begin
19646 :      8871          DODU (.ML_LUN);

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (90)

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Address
19648	:	ML4AD					
19649	:						
19650	:						
19651	:	8872			DOCLN;		
19652	:	8873			end;		
19653	:	8874					
19654	:	8875			ENDTST;		
19658	:						
19659	:						
19663	070772	004167	113614	\$T38:	.SBTTL JSR R1,\$SAVE5		
19664	070776	162706	000012		SUB #12,SP	:	8714
19665	071002	005067	124370		CLR P.CNT	:	
19666	071006	104402		1\$:	TRAP 2	:	8778
19667	071010	152777	000040		BISB #40,@ML.REG+40	:	8780
19668	071016	016705	125010		MOV ML.DUT,R5	:	8781
19669	071022	042705	177770		BIC #177770,R5		
19670	071026	142777	000007	124410	BICB #7,@ML.REG+40		
19671	071034	150577	124404		BISB R5,@ML.REG+40		
19672	071040	005001			CLR R1	:	
19673	071042	012766	000001	000006	MOV #1,6(SP)	:	8783
19674	071050	005004			CLR R4	:	8784
19675	071052	010405		2\$:	MOV R4,R5	:	8786
19676	071054	006305			ASL R5	:	8788
19677	071056	016665	000006	014022	MOV 6(SP),IO.BUF(R5)	:	
19678	071064	005165	014022		COM IO.BUF(R5)	:	
19679	071070	006366	000006		ASL 6(SP)	:	
19680	071074	005204			INC R4	:	8789
19681	071076	020427	000017		CMP R4,#17	:	8786
19682	071102	003763			BLE 2\$:	
19683	071104	004767	127202		JSR PC,GD.BLK.XFER	:	
19684	071110	012777	000061	124266	MOV #61,@ML.REG	:	8792
19685	071116	105777	124332	3\$:	TSTB @ML.REG+50	:	8793
19686	071122	100375			BPL 3\$		
19687	071124	012746	000025		MOV #25,-(SP)	:	
19688	071130	012746	000010		MOV #10,-(SP)	:	8795
19689	071134	004767	125304		JSR PC,STRIPPER		
19690	071140	012702	177777		MOV #-1,R2	:	
19691	071144	012766	000001	000004	MOV #1,4(SP)	:	8796
19692	071152	005066	000010		CLR 10(SP)	:	8797
19693	071156	012766	000003	000006	MOV #3,6(SP)	:	8798
19694	071164	005004			CLR R4	:	8799
19695	071166	005005		4\$:	CLR R5	:	8801
19696	071170	005003		5\$:	CLR R3	:	8804
19697	071172	020366	000010	6\$:	CMP R3,10(SP)	:	8806
19698	071176	001463			BEQ 7\$:	8808
19699	071200	005203			INC R3	:	
19700	071202	005202			INC R2	:	8810
19701	071204	126227	015034	000017	CMPB STACK(R2),#17	:	8811
19702	071212	001767			BEQ 6\$:	8813

Address	Code	Label	Comment	Operation	Operand	Address	Time	Page
19704								
19705								
19706								
19707	071214	005267	124156	INC	P.CNT		29-Mar-1982 16:23:04	TOPS
19708	071220	026767	124152 124152	CMP	P.CNT,LIMIT		29-Mar-1982 16:21:03	PA:<
19709	071226	003047		BGT	7\$			8815
19710	071230	104455		TRAP	55			
19711	071232	000132		.WORD	132			8817
19712	071234	012750		.WORD	SYNC			
19713	071236	026302		.WORD	DUMPER			
19714	071240	012746	010730	MOV	#WRD.19,-(SP)			8818
19715	071244	012746	012216	MOV	#FNC.5,-(SP)			
19716	071250	012746	010650	MOV	#WRD.12,-(SP)			
19717	071254	012746	011676	MOV	#PHR.4,-(SP)			
19718	071260	012746	011140	MOV	#WRD.39,-(SP)			
19719	071264	012746	010766	MOV	#WRD.23,-(SP)			
19720	071270	012746	010322	MOV	#SIX.FMT,-(SP)			
19721	071274	012746	000007	MOV	#7,-(SP)			
19722	071300	010600		MOV	SP,R0			
19723	071302	104414		TRAP	14			
19724	071304	010216		MOV	R2,(SP)			
19725	071306	005046		CLR	-(SP)			8819
19726	071310	116216	015034	MOVB	STACK(R2),(SP)			
19727	071314	012746	177777	MOV	#-1,-(SP)			
19728	071320	012746	006650	MOV	#FMT.5,-(SP)			
19729	071324	012746	000004	MOV	#4,-(SP)			
19730	071330	010600		MOV	SP,R0			
19731	071332	104414		TRAP	14			
19732	071334	012701	000001	MOV	#1,R1			
19733	071340	062706	000030	ADD	#30,SP			
19734	071344	000712		BR	6\$			
19735	071346	005202		INC	R2			
19736	071350	012766	000017 000014	MOV	#17,14(SP)			8820
19737	071356	046666	000004 000014	BIC	4(SP),14(SP)			8815
19738	071364	005000		CLR	R0			8808
19739	071366	156200	015034	BISB	STACK(R2),R0			8825
19740	071372	020066	000014	CMP	R0,14(SP)			8825
19741	071376	001462		BEQ	8\$			
19742	071400	005267	123772	INC	P.CNT			
19743	071404	026767	123766	CMP	P.CNT,LIMIT			8830
19744	071412	003153		BGT	11\$			
19745	071414	104455		TRAP	55			
19746	071416	000133		.WORD	133			8832
19747	071420	012750		.WORD	SYNC			
19748	071422	026302		.WORD	DUMPER			
19749	071424	012746	010730	MOV	#WRD.19,-(SP)			8833
19750	071430	012746	012216	MOV	#FNC.5,-(SP)			
19751	071434	012746	010650	MOV	#WRD.12,-(SP)			
19752	071440	012746	011676	MOV	#PHR.4,-(SP)			
19753	071444	012746	011140	MOV	#WRD.39,-(SP)			
19754	071450	012746	010766	MOV	#WRD.23,-(SP)			
19755	071454	012746	010322	MOV	#SIX.FMT,-(SP)			
19756	071460	012746	000007	MOV	#7,-(SP)			
19757	071464	010600		MOV	SP,R0			
19758	071466	104414		TRAP	14			

Address	OpCode	Operand 1	Operand 2	Operand 3	Comment	Line No.
19760						
19761						
19762						
19763	071470	010216			MOV R2,(SP)	
19764	071472	012746	007370		MOV #FMT.15,-(SP)	: STK.PTR,*
19765	071476	012746	000002		MOV #2,-(SP)	
19766	071502	010600			MOV SP,R0	: SP,*
19767	071504	104414			TRAP 14	
19768	071506	005016			CLR (SP)	
19769	071510	116216	015034		MOVB STACK(R2),(SP)	: *(STK.PTR),*
19770	071514	016646	000040		MOV 40(SP),-(SP)	: SAV.NIB,*
19771	071520	012746	006650		MOV #FMT.5,-(SP)	
19772	071524	012746	000003		MOV #3,-(SP)	
19773	071530	010600			MOV SP,R0	: SP,*
19774	071532	104414			TRAP 14	
19775	071534	012701	000001		MOV #1,R1	: *,DODU.FLG
19776	071540	062706	000032		ADD #32,SP	
19777	071544	005003			CLR R3	: COUNT
19778	071546	020366	000006	8\$:	CMP R3,6(SP)	: COUNT,ALL.ONES.2
19779	071552	001463		9\$:	BEQ 10\$	
19780	071554	005203			INC R3	: COUNT
19781	071556	005202			INC R2	: STK.PTR
19782	071560	126227	015034	000017	CMPB STACK(R2),#17	: *(STK.PTR),*
19783	071566	001767			BEQ 9\$	
19784	071570	005267	123602		INC P.CNT	
19785	071574	026767	123576	123576	CMP P.CNT,LIMIT	:
19786	071602	003047			BGT 10\$	
19787	071604	104455			TRAP 55	:
19788	071606	000134			.WORD 134	:
19789	071610	012750			.WORD SYNC	
19790	071612	026302			.WORD DUMPER	
19791	071614	012746	010730		MOV #WRD.19,-(SP)	:
19792	071620	012746	012216		MOV #FNC.5,-(SP)	:
19793	071624	012746	010650		MOV #WRD.12,-(SP)	
19794	071630	012746	011676		MOV #PHR.4,-(SP)	
19795	071634	012746	011140		MOV #WRD.39,-(SP)	
19796	071640	012746	010766		MOV #WRD.23,-(SP)	
19797	071644	012746	010322		MOV #SIX.FMT,-(SP)	
19798	071650	012746	000007		MOV #7,-(SP)	
19799	071654	010600			MOV SP,R0	: SP,*
19800	071656	104414			TRAP 14	
19801	071660	010216			MOV R2,(SP)	: STK.PTR,*
19802	071662	005046			CLR -(SP)	
19803	071664	116216	015034		MOVB STACK(R2),(SP)	: *(STK.PTR),*
19804	071670	012746	177777		MOV #-1,-(SP)	
19805	071674	012746	006650		MOV #FMT.5,-(SP)	
19806	071700	012746	000004		MOV #4,-(SP)	
19807	071704	010600			MOV SP,R0	: SP,*
19808	071706	104414			TRAP 14	
19809	071710	012701	000001		MOV #1,R1	: *,DODU.FLG
19810	071714	062706	000030		ADD #30,SP	
19811	071720	000712			BR 9\$:
19812	071722	006366	000004	10\$:	ASL 4(SP)	: NIB.BIT
19813	071726	005205			INC R5	: BY.ONE.WRD
19814	071730	020527	000003		CMP R5,#3	: BY.ONE.WRD,*

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

8834

8835

8836

8830

8839

8841

8843

8844

8846

8848

8850

8851

8852

8853

8848

8841

8858

8804

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

```

19816      ;ML4AD
19817      ;
19818      ;
19819 071734 003002      BGT 11$
19820 071736 000167 177226 JMP 5$
19821 071742 012766 000001 000004 11$: MOV #1,4(SP)      ; *.NIB.BIT
19822 071750 005266 000010      INC 10(SP)      ; ALL.ONES.1
19823 071754 005366 000006      DEC 6(SP)      ; ALL.ONES.2
19824 071760 005204      INC R4      ; BY.FOUR.WRDS
19825 071762 020427 000003      CMP R4,#3      ; BY.FOUR.WRDS,*
19826 071766 003002      BGT 12$
19827 071770 000167 177172 JMP 4$
19828 071774 022626      12$: CMP (SP)+,(SP)+      ;
19829 071776 104467      TRAP 67      ;
19830 072000 006000      ROR R0
19831 072002 103002      BHIS 13$
19832 072004 000167 176776 JMP 1$
19833 072010 005301      13$: DEC R1      ; DODU.FLG
19834 072012 001004      BNE 14$
19835 072014 016700 124010      MOV ML.LUN,R0      ;
19836 072020 104451      TRAP 51      ;
19837 072022 104444      TRAP 44
19838 072024 062706 000012      14$: ADD #12,SP      ;
19839 072030 000207      RTS PC      ;
19840
19841      ; Routine Size: 272 words
19842      ; Maximum stack depth per invocation: 26 words
19847
19848
19852
19853      .SBTTL T38 TEST CODE SECTION
19857 072032      T38::
19858 072032 004767 176734 1$: JSR PC,$T38      ;
19859 072036 104466      TRAP 66
19860 072040 006000      ROR R0
19861 072042 103773      BLO 1$
19862 072044 000207      RTS PC
19863
19864      ; Routine Size: 6 words
19865      ; Maximum stack depth per invocation: 0 words
19870 ;      8876 !<BLF/PAGE>

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (91)

19872 :ML4AD
19873 :
19874 :
19875 :
19876 :
19877 :
19878 :
19879 :
19880 :
19881 :
19882 :
19883 :
19884 :
19885 :
19886 :
19887 :
19888 :
19889 :
19890 :
19891 :
19892 :
19893 :
19894 :
19895 :
19896 :
19897 :
19898 :
19899 :
19900 :
19901 :
19902 :
19903 :
19904 :
19905 :
19906 :
19907 :
19908 :
19909 :
19910 :
19911 :
19912 :
19913 :
19914 :
19915 :
19916 :
19917 :
19918 :
19919 :
19920 :
19921 :
19922 :
19923 :
19924 :
19925 :
19926 :

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 39

TEST NAME: SYNC DATA BUS BIT UNIQUENESS TEST/READ PATH

TEST DESCRIPTION:

TEST SYNCHRONOUS DATA BUS READ
PATH 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 INTO THE GOOD BLOCK.
3. CLEAR THE IO_BUF.
4. VIA MBUS READ FUNCTION READ THE SHIFTING PATTERN THROUGH THE READ PATH.
5. INTERIGATE THE IO_BUF FOR THE SHIFTING PATTERN.

IMPLICIT INPUTS:

IO_BUF
A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE FUNCTION ARE FOUND.

--
local

DODU_FLG,
TST_PAT;

!DROP UNIT FLAG
!TEST PATTERN

CLR_MBUS;
DODU_FLG = ZERO;
TST_PAT = ONE;

!ONE IN A FIELD OF ZEROES

incr CNT from 0 to 15 do
begin
IO_BUF [.CNT] = not .TST_PAT;
TST_PAT = .TST_PAT^ONE;
end;

!WRITE 16 WORDS WITH SHIFTED 0 IN A FIELD OF 1'S

GD_BLK_XFER ();
MLCS1 = write;
TIME_OUT_LOOP;
BGNSDB;

!SET UP A GOOD BLOCK XFERR
!WRITE SHIFTING PATTERN

incr CNT from 0 to 15 do
IO_BUF [.CNT] = ZEROES;

!CLEAR OUT THE IO_BUF

CLR_MBUS;


```

20040      :ML4AD
20041      :
20042      :
20043 072372 011346      MOV      (R3),-(SP)
20044 072374 010146      MOV      R1, -(SP)
20045 072376 012746 006506  MOV      #FMT.2, -(SP)
20046 072402 012746 000004  MOV      #4, -(SP)
20047 072406 010600      MOV      SP,R0
20048 072410 104414      TRAP     14      ; SP,*
20049 072412 012705 000001  MOV      #1,R5      ; *,DODU.FLG
20050 072416 062706 000026  ADD      #26,SP
20051 072422 006304      7$: ASL      R4      ; TST.PAT
20052 072424 005202      INC      R2      ; CNT
20053 072426 020227 000017  CMP      R2,#17    ; CNT,*
20054 072432 003713      BLE     6$
20055 072434 104467      TRAP     67
20056 072436 006000      ROR     R0      ;
20057 072440 103647      BLO     3$
20058 072442 005305      DEC     R5      ; DODU.FLG
20059 072444 001004      BNE     8$
20060 072446 016700 123356  MOV     ML.LUN,R0
20061 072452 104451      TRAP     51
20062 072454 104444      TRAP     44
20063 072456 000207      8$: RTS      PC
20064
20065      : Routine Size: 133 words
20066      : Maximum stack depth per invocation: 17 words
20071
20072
20076
20077      .SBTTL T39 TEST CODE SECTION
20081 072460      T39::
20082 072460 004767 177362  1$: JSR     PC,$T39
20083 072464 104466      TRAP     66
20084 072466 006000      ROR     R0
20085 072470 103773      BLO     1$
20086 072472 000207      RTS     PC
20087
20088      : Routine Size: 6 words
20089      : Maximum stack depth per invocation: 0 words
20094 :      8959 !<BLF/PAGE>
  
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (92)

20096 :ML4AD
20097 :
20098 :
20099 :
20100 :
20101 :
20102 :
20103 :
20104 :
20105 :
20106 :
20107 :
20108 :
20109 :
20110 :
20111 :
20112 :
20113 :
20114 :
20115 :
20116 :
20117 :
20118 :
20119 :
20120 :
20121 :
20122 :
20123 :
20124 :
20125 :
20126 :
20127 :
20128 :
20129 :
20130 :
20131 :
20132 :
20133 :
20134 :
20135 :
20136 :
20137 :
20138 :
20139 :
20140 :
20141 :
20142 :
20143 :
20144 :
20145 :
20146 :
20147 :
20148 :
20149 :
20150 :

8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978
8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
8999
9000
9001
9002
9003
9004
9005
9006
9007
9008
9009
9010
9011

TEST CODE SECTION

BGNTST;

!++

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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (92)

```

20152 :ML4AD
20153 :
20154 :
20155 :          9012      DSA_ADRS = .DSA_ADRS + .RAS_INC;
20156 :          9013      MLWC = .W_C_SIZE;
20157 :          9014      MLBA = IO_BUF;
20158 :          9015      MLDA = .DSA_ADRS;
20159 :          9016      ML_FUNC = write;
20160 :          9017      TIME_OUT_LOOP;
20161 :          9018
20162 :          9019      if .SC IS_NOT_SET
20163 :          9020      then
20164 :          9021          begin
20165 :          9022              MLWC = .W_C_SIZE;
20166 :          9023              MLBA = IO_BUF;
20167 :          9024              MLDA = .DSA_ADRS;
20168 :          9025              ML_FUNC = WRT_CHK;
20169 :          9026              TIME_OUT_LOOP;
20170 :          9027
20171 :          9028              if .SC IS_NOT_SET
20172 :          9029              then
20173 :          9030                  FND_GD_CHK = ONE;
20174 :          9031
20175 :          9032              end
20176 :          9033
20177 :          9034          end
20178 :          9035      !
20179 :          9036      !VER CZMLAD CHANGED TEST TO UNSIGNED TEST
20180 :          9037      !
20181 :          9038      until (.FND_GD_CHK IS_SET ) or (.DSA_ADRS eqlU .LST_ARR + .ARR_INC);
20182 :          9039
20183 :          9040              !REPEAT UNTIL FOUND GOOD CHUNCK OR AT LBT
20184 :          9041      !
20185 :          9042      !
20186 :          9043      !VER CZMLAD CHANGED TEST TO UNSIGNED TEST
20187 :          9044      !
20188 :          9045      if .DSA_ADRS eqlU .LST_ARR + .ARR_INC
20189 :          9046      then
20190 :          9047          begin
20191 :          9048              ERRDF (111, INTER, DUMPER);
20192 :          9049              PRINTB (FIV_FMT, FNC_13, FNC_17, WRD_50, WRD_60, WRD_56);
20193 :          9050              PRINTB (THR_FMT, WRD_14, PHR_10, FNC_15);
20194 :          9051          end
20195 :          9052      else
20196 :          9053          begin
20197 :          9054              CLR_MBUS;
20198 :          9055              BAI = ONE;
20199 :          9056              ECC_DIS = ONE;
20200 :          9057              IO_BUF = ZEROES;
20201 :          9058              MLDA = .DSA_ADRS;
20202 :          9059              MLWC = not 255;
20203 :          9060              MLBA = IO_BUF;
20204 :          9061              ML_FUNC = write;
20205 :          9062              TIME_OUT_LOOP;
20206 :          9063              CLR_MBUS;

```

```

!INCREMENT DSA ADRS COUNTER
!16K OR 64K WORDS
!LOAD UBUS ADRS
!LOAD DSA ADRS
!DO A WRITE FUNCTION

!DID XFERR CAUSE AN SC

!XFERR WAS OK
!LOAD WORD COUNT
!LOAD UBUS ADRS
!LOAD DSA ADRS
!DO A WRITE CHECK FUNCTION

!IS THIS CHUNCK GOOD

!YES SET FLG

!REPEAT UNTIL FOUND GOOD CHUNCK OR AT LBT

!A GOOD CHUNCK WAS FOUND CONTINUE TEST

!DISABLE ECC
!FIRST BLOCK IN CHUNCK GETS ZEROES
!LOAD DSA
!LOAD WORD COUNT
!LOAD UBUS ADRS
!DO A WRITE FUNCTION

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (92)

20208 :MI 4AD

TEST CODE SECTION

```

20209 :
20210 :
20211 :          9064      BAI = ONE;
20212 :          9065      IO BUF = ONES;
20213 :          9066      ECC DIS = ONE;
20214 :          9067      MLDA = .DSA ADRS + 1;
20215 :          9068      MLBA = IO_BUF;
20216 :          9069      MLWC = .W_C SIZE + 256;
20217 :          9070      ML_FUNC = WRT_CHK;
20218 :          9071      TIME_OUT_LOOP;
20219 :          9072
20220 :          9073      if .WCE IS_SET
20221 :          9074      then
20222 :          9075          begin
20223 :          9076              ERRDF (112, ASYNC, DUMPER);
20224 :          9077              PRINTB (FOR_FMT, FNC 17, WRD_50, WRD_60, WRD_14);
20225 :          9078              PRINTB (FMT_9, (.MLDA - 1));
20226 :          9079              DODU (.ML_LDN);
20227 :          9080              DOCLN;
20228 :          9081              end;
20229 :          9082
20230 :          9083      end;
20231 :          9084
20232 :          9085      ENDTST;
20236 :
20237 :

```

```

!READ REMAINING BLOCKS FOR ONES
!LOAD DSA WITH NEXT DSA INCREMENT
!LOAD USUS ADRS
!LOAD WORD COUNT WITH LESS ONE BLOCK
!DO A WRITE CHECK FUNCTION

!WERE ANY BITS DISTURBED

!ERROR IF WRITE CHECK FOUND BAD DATA

```

20241	072474	004167	112040		ST40:	.SBTTL	ST40 TEST CODE SECTION		
20242	072500	012767	177777	121314		JSR	R1,SSAVE2	:	8058
20243	072506	016702	122634			MOV	#-1,IO.BUF	:	9003
20244	072512	005402				MOV	RAS.INC,R2	:	9004
20245	072514	005000				NEG	R2	:	
20246	072516	152777	000040	122720	1\$:	CLR	R0	:	9005
20247	072524	016701	123302			BISB	#40,@ML.REG+40	:	9008
20248	072530	042701	177770			MOV	ML,DUT,R1	:	
20249	072534	142777	000007	122702		BIC	#177770,R1	:	
20250	072542	150177	122676			BICB	#7,@ML.REG+40	:	
20251	072546	152777	000010	122670		BISB	R1,@ML.REG+40	:	
20252	072554	152777	000002	122742		BISB	#10,@ML.REG+40	:	9010
20253	072562	066702	122560			BISB	#2,@ML.REG+120	:	9011
20254	072566	016777	122552	122620		ADD	RAS.INC,R2	:	9012
20255	072574	012777	014022	122622		MOV	W.C.SIZE,@ML.REG+10	:	9013
20256	072602	010277	122626			MOV	#10.BUF,@ML.REG+20	:	9014
20257	072606	142777	000077	122570		MOV	R2,@ML.REG+30	:	9015
20258	072614	152777	000061	122562		BICB	#77,@ML.REG	:	9016
20259	072622	105777	122626		2\$:	BISB	#61,@ML.REG	:	
20260	072626	100375				TSTB	@ML.REG+50	:	
20261	072630	032777	100000	122546		BPL	2\$:	
20262	072636	001027				BIT	#100000,@ML.REG	:	9019
						BNE	4\$:	

Address	OpCode	Op1	Op2	Op3	Op4	Comment	Time	Page
20320						:ML4AD	29-Mar-1982 16:23:04	TOPS
20321						:	29-Mar-1982 16:21:03	PA:<
20322						TEST CODE SECTION		
20323	073144	012777	014022	122252		MOV #10,BUF,@ML.REG+20		9060
20324	073152	142777	000077	122224		BICB #77,@ML.REG		9061
20325	073160	152777	000061	122216		BISB #61,@ML.REG		
20326	073166	105777	122262		7\$:	TSTB @ML.REG+50		
20327	073172	100375				BPL 7\$		
20328	073174	152777	000040	122242		BISB #40,@ML.REG+40		9062
20329	073202	016701	122624			MOV ML.DUT,R1		
20330	073206	042701	177770			BIC #177770,R1		
20331	073212	142777	000007	122224		BICB #7,@ML.REG+40		
20332	073220	150177	122220			BISB R1,@ML.REG+40		
20333	073224	152777	000010	122212		BISB #10,@ML.REG+40		9064
20334	073232	012767	177777	120562		MOV #-1,IO.BUF		9065
20335	073240	152777	000002	122256		BISB #2,@ML.REG+120		9066
20336	073246	010201				MOV R2,R1		9067
20337	073250	005201				INC R1	DSA.ADRS,*	
20338	073252	010177	122156			MOV R1,@ML.REG+30		
20339	073256	012777	014022	122140		MOV #10,BUF,@ML.REG+20		9068
20340	073264	016702	122054			MOV W.C.SIZE,R2		9069
20341	073270	062702	000400			ADD #400,R2		
20342	073274	010277	122114			MOV R2,@ML.REG+10		
20343	073300	142777	000077	122076		BICB #77,@ML.REG		9070
20344	073306	152777	000051	122070		BISB #51,@ML.REG		
20345	073314	105777	122134		8\$:	TSTB @ML.REG+50		
20346	073320	100375				BPL 8\$		
20347	073322	032777	040000	122114		BIT #40000,@ML.REG+40		9073
20348	073330	001441				BEQ 9\$		
20349	073332	104455				TRAP 55		9076
20350	073334	000160				.WORD 160		
20351	073336	012706				.WORD ASYNC		
20352	073340	026302				.WORD DUMPER		
20353	073342	012746	010664			MOV #WRD.14,-(SP)		9077
20354	073346	012746	011400			MOV #WRD.60,-(SP)		
20355	073352	012746	011270			MOV #WRD.50,-(SP)		
20356	073356	012746	012404			MOV #FMC.17,-(SP)		
20357	073362	012746	010270			MOV #FC.FMT,-(SP)		
20358	073366	012746	000005			MOV #FC.FMT,-(SP)		
20359	073372	010600				MOV SP,R0	SP,*	
20360	073374	104414				TRAP 14		
20361	073376	017716	122032			MOV @ML.REG+30,(SP)		9078
20362	073402	005316				DEC (SP)		
20363	073404	012746	007064			MOV #FMT.9,-(SP)		
20364	073410	012746	000002			MOV #2,-(SP)		
20365	073414	010600				MOV SP,R0	SP,*	
20366	073416	104414				TRAP 14		
20367	073420	016700	122404			MOV ML.LUN,R0		9079
20368	073424	104451				TRAP 51		
20369	073426	104444				TRAP 44		
20370	073430	062706	000020			ADD #20,SP		9075
20371	073434	000207			9\$:	RTS PC		8958
20372								
20373								
20374								

: Routine Size: 241 words
: Maximum stack depth per invocation: 14 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

20376
20377
20378
20383
20384
20388
20389
20393 073436
20394 073436 004767 177032
20395 073442 104466
20396 073444 006000
20397 073446 103773
20398 073450 000207
20399
20400
20401
20406
20407
20408 :

:ML4AD
:
TEST CODE SECTION

T40:: .SBTTL T40 TEST CODE SECTION
1\$:

JSR PC,\$T40 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

9083

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

9086 !<BLF/PAGE>

20410 :ML4AD
20411 :
20412 :
20413 :
20414 :
20415 :
20416 :
20417 :
20418 :
20419 :
20420 :
20421 :
20422 :
20423 :
20424 :
20425 :
20426 :
20427 :
20428 :
20429 :
20430 :
20431 :
20432 :
20433 :
20434 :
20435 :
20436 :
20437 :
20438 :
20439 :
20440 :
20441 :
20442 :
20443 :
20444 :
20445 :
20446 :
20447 :
20448 :
20449 :
20450 :
20451 :
20452 :
20453 :
20454 :
20455 :
20456 :
20457 :
20458 :
20459 :
20460 :
20461 :
20462 :
20463 :
20464 :

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (93)

9087 !
9088 BGNTST:
9089

9090 !++

9091 TEST NUMBER: TST 41

9092 TEST NAME: NIBBLE OFFSET TEST

9093 TEST DESCRIPTION:

9094 TEST NIBBLE OFFSET COUNTERS TO OFFSET GOOD NIBBLE DATA A MAX
9095 OF 14 WORDS ON DETECTION OF ALL BAD NIBBLES BY:

- 9096 1. LOADING FIRST 2 1/4 WORDS OF THE IO_BUF WITH ZEROES AND THE
9097 REMAINING OF BUFFER WITH ONES.
- 9098 2. VIA DAT_DM MODE WRITE THE GOOD BLOCK WITH BACKGROUND
9099 ON ONES.
- 9100 3. VIA PROM R/W MODE FORCE ALL ARRAY NIBBLES BAD.
- 9101 4. VIA A MBUS WRITE FUNCTION LOAD IO_BUF INTO THE GOOD BLOCK.
- 9102 5. VIA DAT_DM MODE READ FIRST 15 ARRAY WORDS FOR ZEROES AND THE
9103 REMAINING WORDS FOR ONES.

9104 IMPLICIT INPUTS:

9105 PD_TEMP
9106 A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED AND
9107 ACCESSED FROM.

9108 IO_BUF
9109 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
9110 FUNCTIONS ARE FOUND.

9111 --
9112 local

9113 DODU_FLG,
9114 TST_PAT,
9115 START,
9116 FINISH,
9117 ERR_FLG;

!DROP UNIT FLAG
!TEST PATTERN
!STARTING WORD
!ENDING WORD
!ERROR FLAG

9118 CLR_THRESHOLD;

!CLEAR ERROR PRINT THRESHOLD

9119 BGNSUB;

9120 CLR_MBUS;

9121 DODU_FLG = ZERO;

9122 incr WD_CNT from 0 to 255 do

9123 IO_BUF [WD_CNT] = ONES;

!LOAD IO_BUF WITH ONES

9124 IO_BUF [0] = ZEROES;

9125 IO_BUF [1] = ZEROES;

!LOAD FIRST 2 1/4 WORDS WITH ZEROES

```

20466 :ML4AD
20467 :
20468 :
20469 : 9139 IO BUF [2] = %o'177760';
20470 : 9140 MLD1 = ONES;
20471 : 9141 MLD2 = ONES;
20472 : 9142 MLE2 = ONES;
20473 : 9143 DAT_DM_XFER ();
20474 : 9144 MLC51 = write;
20475 : 9145
20476 : 9146 incr WD_CNT from 0 to 127 do
20477 : 9147   begin
20478 : 9148     DELAY (ONE_US);
20479 : 9149     DAT_CLK = ONE;
20480 : 9150   end;
20481 : 9151
20482 : 9152 CLR_MBUS;
20483 : 9153 WRT_PD (ONES, 19);
20484 : 9154 PROM_RW = ONE;
20485 : 9155 GD_BLK_XFER ();
20486 : 9156 MLC51 = write;
20487 : 9157
20488 : 9158 TIME_OUT_LOOP;
20489 : 9159 CLR_MBUS;
20490 : 9160 START = ZERO;
20491 : 9161 FINISH = 14;
20492 : 9162 TST_PAT = ZEROES;
20493 : 9163 DAT_DM_XFER ();
20494 : 9164 MLC51 = read;
20495 : 9165 DELAY (ONE_US);
20496 : 9166
20497 : 9167 incr TWICE from 0 to 1 do
20498 : 9168   begin
20499 : 9169
20500 : 9170   incr WRD_CNT from .START to .FINISH do
20501 : 9171     begin
20502 : 9172       PD_TEMP = .MLPD;
20503 : 9173       DAT_CLK = ONE;
20504 : 9174       DELAY (ONE_US);
20505 : 9175       RD_LNG_WRD;
20506 : 9176
20507 : 9177       incr NIB_PTR from 0 to 8 do
20508 : 9178
20509 : 9179         if .PD_TEMP [.NIB_PTR] IS_NOT_SET
20510 : 9180         then
20511 : 9181           begin
20512 : 9182             TST_LNG_WRD (.NIB_PTR, .TST_PAT, ERR_FLG);
20513 : 9183
20514 : 9184             if .ERR_FLG IS_SET
20515 : 9185             then
20516 : 9186               begin
20517 : 9187                 CMP_THRESHOLD;
20518 : 9188                 ERRDF (94, ARR_DAT, DUMPER);
20519 : 9189                 PRINTB (THR_FMT, WRD_41, WRD_46, WRD_10);
20520 : 9190                 PRINTB (FMT_6, .NIB_PTR);

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (93)

!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
!COMPARE ERROR PRINT THRESHOLD

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (93)

```

20522 :ML4AD
20523 :
20524 :
20525 : 9191          DODU_FLG = ONE;
20526 : 9192          end;
20527 : 9193
20528 : 9194          end;
20529 : 9195
20530 : 9196          end;
20531 : 9197
20532 : 9198          TST PAT = not .TST_PAT;
20533 : 9199          START = 15;
20534 : 9200          FINISH = 126;
20535 : 9201          end
20536 : 9202
20537 : 9203          ENDSUB;
20538 : 9204
20539 : 9205          if .DODU_FLG IS_SET
20540 : 9206          then
20541 : 9207              begin
20542 : 9208                  DODU (.ML_LUN);
20543 : 9209                  DOCLN;
20544 : 9210              end;
20545 : 9211
20546 : 9212          ENDTST;

```

!NOW READ FOR ONES
!START A 15
!END AT 126

!DROP THIS UNIT IF DODU_FLG SET

Address	Hex	Dec	Label	OpCode	Comment	Address
20551				.SBTTL	\$T41 TEST CODE SECTION	
20555	073452	004167	111134	\$T41: JSR	R1,\$SAVE5	9085
20556	073456	162706	000012	SUB	#12,SP	
20557	073462	005067	121710	CLR	P.CNT	9127
20558	073466	104402		1\$: TRAP	2	9129
20559	073470	152777	000040	121746	BISB #40,@ML.REG+40	9130
20560	073476	016704	122330	MOV	ML,DUT,R4	
20561	073502	042704	177770	BIC	#177770,R4	
20562	073506	142777	000007	121730	BICB #7,@ML.REG+40	
20563	073514	150477	121724	BISB	R4,@ML.REG+40	
20564	073520	005066	000004	CLR	4(SP)	: DODU.FLG 9132
20565	073524	005002		CLR	R2	: WD.CNT 9134
20566	073526	010203		2\$: MOV	R2,R3	: WD.CNT,+ 9135
20567	073530	006303		ASL	R3	
20568	073532	012763	177777	014022	MOV #-1,IO.BUF(R3)	
20569	073540	005202		INC	R2	: WD.CNT 9134
20570	073542	020227	000377	CMP	R2,#377	: WD.CNT,+
20571	073546	003767		BLE	2\$	
20572	073550	005067	120246	CLR	IO.BUF	: 9137
20573	073554	005067	120244	CLR	IO.BUF+2	: 9138
20574	073560	012767	177760	120240	MOV #-20,IO.BUF+4	: 9139
20575	073566	012777	177777	122000	MOV #-1,@ML.REG+170	: 9140
20576	073574	012777	177777	122002	MOV #-1,@ML.REG+200	: 9141

Address	OpCode	Op1	Op2	Op3	Op4	Instruction	Comments	Line No.
20578								
20579								
20580								
20581	073602	012777	177777	121754		MOV #1,AML.REG+160		
20582	073610	004767	122576			JSR PC,DAT.DM.XFER		9142
20583	073614	012777	000061	121562		MOV #61,AML.REG		9143
20584	073622	005001				CLR R1	WD.CNT	9144
20585	073624	012702	000001		3\$:	MOV #1,R2	*,\$STMP2	9146
20586	073630	001411			4\$:	BEQ 7\$		9148
20587	073632	016703	106260			MOV LSDLY,R3	*,\$STMP1	
20588	073636	001404				BEQ 6\$		
20589	073640	005066	000010		5\$:	CLR 10(SP)	\$STMP	
20590	073644	005303				DEC R3	\$STMP1	
20591	073646	001374				BNE 5\$		
20592	073650	005302			6\$:	DEC R2	\$STMP2	
20593	073652	000766				BR 4\$		
20594	073654	157777	000020	121642	7\$:	BISB #20,AML.REG+120		9149
20595	073662	005201				INC R1	WD.CNT	9146
20596	073664	020127	000177			CMP R1,#177	WD.CNT,*	
20597	073670	003755				BLE 3\$		
20598	073672	152777	000040	121544		BISB #40,AML.REG+40		9150
20599	073700	016704	122126			MOV ML,DUT,R4		
20600	073704	042704	177770			BIC #177770,R4		
20601	073710	142777	000007	121526		BICB #7,AML.REG+40		
20602	073716	150477	121522			BISB R4,AML.REG+40		
20603	073722	012746	177777			MOV #-1,-(SP)		
20604	073726	012746	000023			MOV #23,-(SP)		9153
20605	073732	004767	127512			JSR PC,WRT.PD		
20606	073736	152777	000100	121560		BISB #100,AML.REG+120		9154
20607	073744	004767	124342			JSR PC,GD.BLK.XFER		9155
20608	073750	012777	000061	121426		MOV #61,AML.REG		9156
20609	073756	105777	121472		8\$:	TSTB AML.REG+50		
20610	073762	100375				BPL 8\$		
20611	073764	152777	000040	121452		BISB #40,AML.REG+40		9158
20612	073772	016704	122034			MOV ML,DUT,R4		
20613	073776	042704	177770			BIC #177770,R4		
20614	074002	142777	000007	121434		BICB #7,AML.REG+40		
20615	074010	150477	121430			BISB R4,AML.REG+40		
20616	074014	005066	000006			CLR 6(SP)	START	9160
20617	074020	012766	000016	000004		MOV #16,4(SP)	*FINISH	9161
20618	074026	005005				CLR R5	TST.PAT	9162
20619	074030	004767	122356			JSR PC,DAT.DM.XFER		9163
20620	074034	012777	000071	121342		MOV #71,AML.REG		9164
20621	074042	012702	000001			MOV #1,R2	*,\$STMP2	9165
20622	074046	001411			9\$:	BEQ 12\$		
20623	074050	016703	106042			MOV LSDLY,R3	*,\$STMP1	
20624	074054	001404				BEQ 11\$		
20625	074056	005066	000014		10\$:	CLR 14(SP)	\$STMP	
20626	074062	005303				DEC R3	\$STMP1	
20627	074064	001374				BNE 10\$		
20628	074066	005302			11\$:	DEC R2	\$STMP2	
20629	074070	000766				BR 9\$		
20630	074072	005004			12\$:	CLR R4	TWICE	9167
20631	074074	016601	000006		13\$:	MOV 6(SP),R1	START,WRD.CNT	9170
20632	074100	005301				DEC R1	WRD.CNT	

Address	OpCode	Operand 1	Operand 2	Operand 3	Label	Instruction	Comments	Address
20690								
20691								
20692								
20693	074350	104414				TRAP 14		
20694	074352	010216				MOV R2,(SP)	: NIB.PTR,*	9190
20695	074354	012746	006752			MOV #FMT.6,-(SP)		
20696	074360	012746	000002			MOV #2,-(SP)		
20697	074364	G10600				MOV SP,R0	: SP,*	
20698	074366	104414				TRAP 14		
20699	074370	012766	000001	000034		MOV #1,34(SP)	: *,DODU.FLG	9191
20700	074376	062706	000016			ADD #16,SP	:	9186
20701	074402	062706	000006		21\$:	ADD #6,SP	:	9181
20702	074406	005202			22\$:	INC R2	: NIB.PTR	9177
20703	074410	020227	000010			CMP R2,#10	: NIB.PTR,*	
20704	074414	G03667				BLE 19\$		
20705	074416	005201			23\$:	INC R1	: WRD.CNT	9170
20706	074420	020166	000004			CMP R1,4(SP)	: WRD.CNT,FINISH	
20707	074424	003627				BLE 14\$		
20708	074426	005105				COM R5	: TST.PAT	9198
20709	074430	012766	000017	000006		MOV #17,6(SP)	: *,START	9199
20710	074436	012766	000176	000004		MOV #176,4(SP)	: *,FINISH	9200
20711	074444	005204				INC R4	: TWICE	9167
20712	074446	020427	000001			CMP R4,#1	: TWICE,*	
20713	074452	003610				BLE 13\$		
20714	074454	022626				CMP (SP)+,(SP)+	:	9129
20715	074456	104467				TRAP 67	:	9201
20716	074460	006000				ROR R0		
20717	074462	103002				BHIS 24\$		
20718	074464	000167	176776			JMP 1\$		
20719	074470	026627	000004	000001	24\$:	CMP 4(SP),#1	: DODU.FLG,*	9205
20720	074476	001004				BNE 25\$		
20721	074500	016700	121324			MOV ML.LUN,R0	:	9208
20722	074504	104451				TRAP 51		
20723	074506	104444				TRAP 44		
20724	074510	062706	000012		25\$:	ADD #12,SP	:	9085
20725	074514	000207				RTS PC		
20726								
20727								
20728								
20733								
20734								
20738								
20739								
20743	074516				T41::	.SBTTL T41 TEST CODE SECTION		
20744	074516	004767	176730		1\$:	JSR PC,ST41	:	9210

: Routine Size: 274 words
 : Maximum stack depth per invocation: 23 words

20746
20747
20748
20749 074522 104466
20750 074524 006000
20751 074526 103773
20752 074530 000207
20753
20754
20755
20760
20761
20762 :

:ML4AD
:
TEST CODE SECTION

29-Mar-1982 16:23.04 TOPS
29-Mar-1982 16:21:03 PA:<

TRAP 66
ROR R0
BLO 1\$
RTS PC

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

9213 !<BLF/PAGE>

20764 :ML4AD
20765 :
20766 :
20767 :
20768 :
20769 :
20770 :
20771 :
20772 :
20773 :
20774 :
20775 :
20776 :
20777 :
20778 :
20779 :
20780 :
20781 :
20782 :
20783 :
20784 :
20785 :
20786 :
20787 :
20788 :
20789 :
20790 :
20791 :
20792 :
20793 :
20794 :
20795 :
20796 :
20797 :
20798 :
20799 :
20800 :
20801 :
20802 :
20803 :
20804 :
20805 :
20806 :
20807 :
20808 :
20809 :
20810 :
20811 :
20812 :
20813 :
20814 :
20815 :
20816 :
20817 :
20818 :

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (94)

9214 !
9215 !
9216 !
9217 !
9218 !
9219 !
9220 !
9221 !
9222 !
9223 !
9224 !
9225 !
9226 !
9227 !
9228 !
9229 !
9230 !
9231 !
9232 !
9233 !
9234 !
9235 !
9236 !
9237 !
9238 !
9239 !
9240 !
9241 !
9242 !
9243 !
9244 !
9245 !
9246 !
9247 !
9248 !
9249 !
9250 !
9251 !
9252 !
9253 !
9254 !
9255 !
9256 !
9257 !
9258 !
9259 !
9260 !
9261 !
9262 !
9263 !
9264 !
9265 !

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_THRESHOLD;

!CLEAR ERROR PRINT THRESHOLD

CLR_MBUS;

BAI = ONE;

!SET ON FIRST IO_BUF ADRS

IO_BUF = ONES;

!LOAD FIRST IO_BUF ADRS

GD_BLK_XFER ();

!SET UP A GOOD BLOCK XFERR

MLCS1 = write;

!WRITE BACKGROUND PATTERN

TIME_OUT_LOOP;

incr TWICE from 0 to 1 do

!FORCE CLASS 'A' AND CLASS 'B' ERRORS

begin

BGNSUB;

CLR_MBUS;

incr CNT from 0 to 255 do

!CLEAR OUT IO_BUF

IO_BUF [CNT] = ZEROES;

if .TWICE eql 1

!IF 2ND PASS THEN FORCE 'B' ERROR

then

begin

PROM_RW = ONE;

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (94)

20820 :ML4AD
20821 :
20822 :
20823 :
20824 :
20825 :
20826 :
20827 :
20828 :
20829 :
20830 :
20831 :
20832 :
20833 :
20834 :
20835 :
20836 :
20837 :
20838 :
20839 :
20840 :
20841 :
20842 :
20843 :
20844 :
20845 :
20846 :
20847 :
20848 :
20849 :
20850 :
20851 :
20852 :
20853 :
20854 :
20855 :
20856 :
20857 :
20858 :
20859 :
20860 :
20861 :
20862 :
20863 :
20864 :
20865 :
20866 :
20867 :
20868 :
20869 :
20870 :
20871 :
20872 :
20873 :
20874 :

TEST CODE SECTION

```

    MLPD = %o'777';
end;

GD_BLK_XFER ();
MLCSI = 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
    CMP THRESHOLD;
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
    CMP THRESHOLD;
ERRDF (96, SYNC, DUMPER);
PRINTB (FOR_FMT, PHR_9, FNC_13, WRD_19, WRD_10);
exitloop;
end;
end;
end;
end;

if .SC IS_NOT_SET
then
begin
ERRDF (104, SYNC, DUMPER);

```

```

!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
!COMPARE ERROR PRINT THRESHOLD

!CLASS 'B' ERROR
!SEE IF XFERR WAS ABORTED
!READ IO_BUF CLEARED DATA
!ERROR IF ONES AND EXIT LOOP
!COMPARE ERROR PRINT THRESHOLD

!SEE IF SC BIT SET
!ERROR IF NOT SET

```

20876 :ML4AD
20877 :
20878 :
20879 :
20880 :
20881 :
20882 :
20883 :
20884 :
20885 :
20889 :
20890 :

TEST CODE SECTION

PRINTB (FIV_FMT, WRD_59, PHR_1, WRD_11, WRD_19, FNC_13)
end;

ENDSUB;
end;

ENDTST;

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (94)

Address	Hex	Hex	Hex	Label	Instruction	Comment	Address	
20894	074532	004167	110016	\$T42:	.SBTTL	\$T42 TEST CODE SECTION		
20895	074536	005067	120634		JSR	R1,\$SAVE3	9212	
20896	074542	152777	000040	120674	CLR	P,CNT	9215	
20897	074550	016703	121256		BISB	#40,AML.REG+40	9246	
20898	074554	042703	177770		MOV	ML,DUT,R3		
20899	074560	142777	000007	120656	BIC	#177770,R3		
20900	074566	150377	120652		BICB	#7,AML.REG+40		
20901	074572	152777	000010	120644	BISB	R3,AML.REG+40		
20902	074600	012767	177777	117214	BISB	#10,AML.REG+40	9248	
20903	074606	004767	123500		MOV	#-1,IO.BUF	9249	
20904	074612	012777	000061	120564	JSR	PC,GD.BLK.XFER	9250	
20905	074620	105777	120630	1\$:	MOV	#61,AML.REG	9251	
20906	074624	100375			TSTB	AML.REG+50		
20907	074626	005003			BPL	1\$		
20908	074630	104402		2\$:	CLR	R3	: TWICE	
20909	074632	152777	000040	120604	TRAP	2	9254	
20910	074640	016702	121166		BISB	#40,AML.REG+40	9255	
20911	074644	042702	177770		MOV	ML,DUT,R2	9256	
20912	074650	142777	000007	120566	BIC	#177770,R2		
20913	074656	150277	120562		BICB	#7,AML.REG+40		
20914	074662	005000			BISB	R2,AML.REG+40		
20915	074664	010001		3\$:	CLR	R0	: CNT	
20916	074666	006301			MOV	R0,R1	: CNT,*	
20917	074670	005061	014022		ASL	R1		
20918	074674	005200			CLR	IO.BUF(R1)		
20919	074676	020027	000377		INC	R0	: CNT	
20920	074702	003770			CMP	R0,#377	: CNT,*	
20921	074704	020327	000001		BLE	3\$		
20922	074710	001006			CMP	R3,#1	: TWICE,*	
20923	074712	152777	000100	120604	BNE	4\$	9262	
20924	074720	012777	000777	120706	BISB	#100,AML.REG+120	9265	
20925	074726	004767	123360		MOV	#777,AML.REG+230	9266	
20926	074732	012777	000071	120444	4\$:	JSR	PC,GD.BLK.XFER	9269
20927	074740	005002			MOV	#71,AML.REG	9270	
20928	074742	005703			CLR	R2	9272	
20929	074744	001004			TST	R3	: TWICE	
20930	074746	005202			BNE	5\$		
					INC	R2		

Address	Hex	Hex	Hex	Hex	Label	Instruction	Comment	Address
20932					:ML4AD			
20933					:	TEST CODE SECTION		
20934								
20935	074750	012777	177777	120456	5\$:	MOV # -1, @ML.REG+30		
20936	074756	105777	120472			TSTB @ML.REG+50		
20937	074762	100375				BPL 5\$		
20938	074764	006002				ROR R2	:	9276
20939	074766	103045				BCC 8\$		
20940	074770	005002				CLR R2	:	9280
20941	074772	010201			6\$:	MOV R2, R1	:	9283
20942	074774	006301				ASL R1	:	
20943	074776	026127	014022	177777		CMP IO.BUF(R1), # -1		
20944	075004	001431				BEQ 7\$		
20945	075006	005267	120364			INC P.CNT	:	9285
20946	075012	026767	120360	120360		CMP P.CNT, LIMIT		
20947	075020	003075				BGT 12\$		
20948	075022	104455				TRAP 5\$:	9287
20949	075024	000137				.WORD 137		
20950	075026	012750				.WORD SYNC		
20951	075030	026302				.WORD DUMPER		
20952	075032	012746	010630			MOV #WRD.10, -(SP)	:	9288
20953	075036	012746	010730			MOV #WRD.19, -(SP)		
20954	075042	012746	012334			MOV #FNC.13, -(SP)		
20955	075046	012746	011762			MOV #PHR.8, -(SP)		
20956	075052	012746	010270			MOV #FOR.FMT, -(SP)		
20957	075056	012746	000005			MOV #5, -(SP)		
20958	075062	010600				MOV SP, R0	:	SP, *
20959	075064	104414				TRAP 14		
20960	075066	000443				BR 10\$:	9289
20961	075070	005202			7\$:	INC R2	:	9280
20962	075072	020227	000100			CMP R2, #100	:	
20963	075076	003735				BLE 6\$:	WRD.CNT, *
20964	075100	000445				BR 12\$:	9276
20965	075102	005002			8\$:	CLR R2	:	9298
20966	075104	010201			9\$:	MOV R2, R1	:	9301
20967	075106	006301				ASL R1		
20968	075110	005761	014022			TST IO.BUF(R1)		
20969	075114	001433				BEQ 11\$		
20970	075116	005267	120254			INC P.CNT	:	9303
20971	075122	026767	120250	120250		CMP P.CNT, LIMIT		
20972	075130	003031				BGT 12\$		
20973	075132	104455				TRAP 5\$:	9305
20974	075134	000140				.WORD 140		
20975	075136	012750				.WORD SYNC		
20976	075140	026302				.WORD DUMPER		
20977	075142	012746	010630			MOV #WRD.10, -(SP)	:	9306
20978	075146	012746	010730			MOV #WRD.19, -(SP)		
20979	075152	012746	012334			MOV #FNC.13, -(SP)		
20980	075156	012746	011774			MOV #PHR.9, -(SP)		
20981	075162	012746	010270			MOV #FOR.FMT, -(SP)		
20982	075166	012746	000005			MOV #5, -(SP)		
20983	075172	010600				MOV SP, R0	:	SP, *
20984	075174	104414				TRAP 14		
20985	075176	062706	000014		10\$:	ADD #14, SP	:	9307
20986	075202	000404				BR 12\$		

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA: <

Address	Hex	Hex	Hex	Label	Code	Comment	Count	Time	Page
20988				:ML4AD				29-Mar-1982 16:23:04	TOPS
20989				:	TEST CODE SECTION			29-Mar-1982 16:21:03	PA:<
20990				:					
20991	075204	005202		11\$:	INC R2		: WRD.CNT		9298
20992	075206	020227	000100		CMP R2,#100		: WRD.CNT,*		
20993	075212	003734			BLE 9\$				
20994	075214	032777	100000	12\$:	BIT #100000,@ML.REG				9314
20995	075222	001026			BNE 13\$				
20996	075224	104455			TRAP 5\$				9317
20997	075226	000150			.WORD 150				
20998	075230	012750			.WORD SYNC				
20999	075232	026302			.WORD DUMPER				
21000	075234	012746	012334		MOV #FNC.13,-(SP)				9318
21001	075240	012746	010730		MOV #WRD.19,-(SP)				
21002	075244	012746	010640		MOV #WRD.11,-(SP)				
21003	075250	012746	011610		MOV #PHR.1,-(SP)				
21004	075254	012746	011374		MOV #WRD.59,-(SP)				
21005	075260	012746	010304		MOV #FIV.FMT,-(SP)				
21006	075264	012746	000006		MOV #6,-(SP)				
21007	075270	010600			MOV SP,R0		: SP,*		
21008	075272	104414			TRAP 14				
21009	075274	062706	000016		ADD #16,SP				9316
21010	075300	104467		13\$:	TRAP 67				9319
21011	075302	006000			ROR R0				
21012	075304	103002			BHIS 15\$				
21013	075306	000167	177316	14\$:	JMP 2\$				
21014	075312	005203		15\$:	INC R3		: TWICE		9254
21015	075314	020327	000001		CMP R3,#1		: TWICE,*		
21016	075320	003772			BLE 14\$				
21017	075322	000207			RTS PC				9212
21018									
21019					: Routine Size: 189 words				
21020					: Maximum stack depth per invocation: 11 words				
21025									
21026									
21030									
21031					.SBTTL T42 TEST CODE SECTION				
21035	075324			T42::					
21036	075324	004767	177202	1\$:	JSR PC,\$T42				9322
21037	075330	104466			TRAP 66				
21038	075332	006000			ROR R0				
21039	075334	103773			BLO 1\$				
21040	075336	000207			RTS PC				
21041									
21042					: Routine Size: 6 words				

21044
21045
21046
21047
21052
21053
21054 ;

:ML4AD

TEST CODE SECTION

: Maximum stack depth per invocation: 0 words

9325 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

21056 :ML4AD
21057 :
21058 :
21059 :
21060 :
21061 :
21062 :
21063 :
21064 :
21065 :
21066 :
21067 :
21068 :
21069 :
21070 :
21071 :
21072 :
21073 :
21074 :
21075 :
21076 :
21077 :
21078 :
21079 :
21080 :
21081 :
21082 :
21083 :
21084 :
21085 :
21086 :
21087 :
21088 :
21089 :
21090 :
21091 :
21092 :
21093 :
21094 :
21095 :
21096 :
21097 :
21098 :
21099 :
21100 :
21101 :
21102 :
21103 :
21104 :
21105 :
21106 :
21107 :
21108 :
21109 :
21110 :

9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377

TEST CODE SECTION

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.

CLR_THRESHOLD;

!CLEAR ERROR PRINT THRESHOLD

!VERSION CZMLAD CHANGED TEST TO UNSIGNED TEST

incrU DSA_CNT from 0 to .LST_BLK - 1 do

!DO XFERRS UP TO THE LAST BLOCK

begin

BGNSUB;

CLR_MBUS;

ECC_DIS = ONE;

MLWC = not 255;

MLBA = IO BUF;

MLDA = .DSA_CNT;

MLCS1 = write;

TIME_OUT_LOOP;

if .LBT IS_SET

!DISABLE ECC
!LOAD WORD COUNT
!LOAD MBUS ADRS
!LOAD DSA WITH DSA CNT
!DO A WRITE FUNCTION

!SEE IF THE LAST BLOCK XFERR BIT SET

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (95)

```

21112 :ML4AD
21113 :
21114 :
21115 : 9378 then
21116 : 9379 begin
21117 : 9380 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
21118 : 9381 ERRDF (97, ASYNC, DUMPER);
21119 : 9382 PRINTB (THR_FMT, WRD_27, PHR_5, WRD_29);
21120 : 9383 PRINTB (FMT_7, .DSA_CNT);
21121 : 9384 end;
21122 : 9385
21123 : 9386 if .MLDA neq .DSA_CNT + 1 !SEE IF THE DSA REG INCREMENTED
21124 : 9387 then
21125 : 9388 begin
21126 : 9389 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
21127 : 9390 ERRDF (98, ASYNC, DUMPER);
21128 : 9391 PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
21129 : 9392 PRINTB (FMT_7, .DSA_CNT);
21130 : 9393 end;
21131 : 9394
21132 : 9395 ENDSUB;
21133 : 9396 end;
21134 : 9397
21135 : 9398 BGNSUB;
21136 : 9399 CLR_MBUS;
21137 : 9400 ECC_DIS = ONE; !DISABLE ECC
21138 : 9401 LAST_BLK_XFER (); !SET UP A LAST BLOCK XFERR
21139 : 9402 MLCST = write; !DO A WRITE FUNCTION
21140 : 9403 TIME_OUT_LOOP;
21141 : 9404
21142 : 9405 if .MLDA neq .LST_BLK + 1 !SEE IF DSA REGISTER INCREMENTED
21143 : 9406 then
21144 : 9407 begin
21145 : 9408 ERRDF (101, ASYNC, DUMPER);
21146 : 9409 PRINTB (THR_FMT, REG_6, WRD_31, WRD_14);
21147 : 9410 PRINTB (FMT_7, .LST_BLK);
21148 : 9411 end;
21149 : 9412
21150 : 9413 if .LBT IS_SET !SEE IF LBT BIT SET
21151 : 9414 then
21152 : 9415 begin
21153 : 9416 MLDA = ONE; !IF SET THEN TRY TO CLEAR IT
21154 : 9417
21155 : 9418 if .LBT IS_SET !SEE IF BIT CLEARED
21156 : 9419 then
21157 : 9420 begin !ERROR IF NOT
21158 : 9421 ERRDF (99, ASYNC, DUMPER);
21159 : 9422 PRINTB (FIV_FMT, WRD_27, PHR_2, WRD_11, WRD_17, REG_6);
21160 : 9423 end;
21161 : 9424
21162 : 9425 end
21163 : 9426 else !LBT NOT SET
21164 : 9427 begin
21165 : 9428 ERRDF (100, ASYNC, DUMPER);
21166 : 9429 PRINTB (FOR_FMT, WRD_27, PHR_1, WRD_11, WRD_27);

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (95)

21168 :ML4AD
21169 :
21170 :
21171 : 9430 end:
21172 : 9431
21173 : 9432 ENDSUB:
21174 : 9433 ENDTST:
21178 :
21179 :

29-Mar-1982 16:23:04 TOPs-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (95)

21183	075340	004167	107210		.SBTTL	\$T43 TEST CODE SECTION		
21184	075344	005067	120026	\$T43:	JSR	R1,\$SAVE3	:	9324
21185	075350	C16703	116440		CLR	P.CNT	:	9328
21186	075354	005303			MOV	LST.BLK,R3	:	9366
21187	075356	005001			DEC	R3	:	
21188	075360	000551			CLR	R1	:	DSA.CNT
21189	075362	010102		1\$:	BR	7\$:	
21190	075364	005202			MOV	R1,R2	:	DSA.CNT,*
21191	075366	104402		2\$:	INC	R2	:	9386
21192	075370	152777	000040	120046	TRAP	2	:	9367
21193	075376	016700	120430		BISB	#40,@ML.REG+40	:	9368
21194	075402	042700	177770		MOV	ML.DUT,R0	:	
21195	075406	142777	000007	120030	BIC	#177770,R0	:	
21196	075414	150077	120024		BICB	#7,@ML.REG+40	:	
21197	075420	152777	000002	120076	BISB	R0,@ML.REG+40	:	
21198	075426	012777	177400	117760	BISB	#2,@ML.REG+120	:	9370
21199	075434	012777	014022	117762	MOV	#-400,@ML.REG+10	:	9371
21200	075442	010177	117766		MOV	#10.BUF,@ML.REG+20	:	9372
21201	075446	012777	000061	117730	MOV	R1,@ML.REG+30	:	DSA.CNT,*
21202	075454	105777	117774		MOV	#61,@ML.REG	:	9373
21203	075460	100375		3\$:	TSTB	@ML.REG+50	:	9374
21204	075462	032777	002000	117764	BPL	3\$:	
21205	075470	001437			BIT	#2000,@ML.REG+50	:	9377
21206	075472	005267	117700		BEQ	4\$:	
21207	075476	026767	117674	117674	INC	P.CNT	:	9379
21208	075504	003076			CMP	P.CNT,LIMIT	:	
21209	075506	104455			BGT	6\$:	
21210	075510	000141			TRAP	55	:	9381
21211	075512	012706			.WORD	141	:	
21212	075514	026302			.WORD	ASYNCR	:	
21213	075516	012746	011032		.WORD	DUMPER	:	
21214	075522	012746	011714		MOV	#WRD.29,-(SP)	:	9382
21215	075526	012746	011024		MOV	#PHR.5,-(SP)	:	
21216	075532	012746	010256		MOV	#WRD.27,-(SP)	:	
21217	075536	012746	000004		MOV	#THR.FMT,-(SP)	:	
21218	075542	010600			MOV	#4,-(SP)	:	
21219	075544	104414			MOV	SP,R0	:	SP,*
21220	075546	010116			TRAP	14	:	
21221	075550	012746	007002		MOV	R1,(SP)	:	DSA.CNT,*
21222	075554	012746	000002		MOV	#FMT.7,-(SP)	:	9383
					MOV	#2,-(SP)	:	

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Address
21224							
21225							
21226							
21227	075560	010600			MOV SP,R0	: SP,*	
21228	075562	104414			TRAP 14		
21229	075564	062706	000016		ADD #16,SP		
21230	075570	027702	117640	4\$:	CMP @ML.REG+30,R2		9379
21231	075574	001437			BEQ 5\$		9386
21232	075576	005267	117574		INC P.CNT		
21233	075602	026767	117570	117570	CMP P.CNT,LIMIT		9388
21234	075610	003034			BGT 6\$		
21235	075612	104455			TRAP 55		9390
21236	075614	000142			.WORD 142		
21237	075614	012706			.WORD ASYNC		
21238	075620	026302			.WORD DUMPER		
21239	075622	012746	010664		MOV #WRD.14,-(SP)		9391
21240	075626	012746	011054		MOV #WRD.31,-(SP)		
21241	075632	012746	012540		MOV #REG.6,-(SP)		
21242	075636	012746	010256		MOV #THR.FMT,-(SP)		
21243	075642	012746	000004		MOV #4,-(SP)		
21244	075646	010600			MOV SP,R0	: SP,*	
21245	075650	104414			TRAP 14		
21246	075652	010116			MOV R1,(SP)	: DSA.CNT,*	9392
21247	075654	012746	007002		MOV #FMT.7,-(SP)		
21248	075660	012746	000002		MOV #2,-(SP)		
21249	075664	010600			MOV SP,R0	: SP,*	
21250	075666	104414			TRAP 14		
21251	075670	062706	000016		ADD #16,SP		9388
21252	075674	104467		5\$:	TRAP 67		9393
21253	075676	006000			ROR R0		
21254	075700	103632			BLO 2\$		
21255	075702	005201		6\$:	INC R1	: DSA.CNT	9366
21256	075704	020103		7\$:	CMP R1,R3	: DSA.CNT,*	
21257	075706	101625			BLOS 1\$		
21258	075710	104402		8\$:	TRAP 2		9396
21259	075712	152777	000040	117524	BISB #40,@ML.REG+40		9398
21260	075720	016703	120106		MOV ML.DUT,R3		
21261	075724	042703	177770		BIC #177770,R3		
21262	075730	142777	000007	117506	BICB #7,@ML.REG+40		
21263	075736	150377	117502		BISB R3,@ML.REG+40		
21264	075742	152777	000002	117554	BISB #2,@ML.REG+120		
21265	075750	004767	122370		JSR PC,LAST.BLK.XFER		9400
21266	075754	012777	000061	117422	MOV #61,@ML.REG		9401
21267	075762	105777	117466	9\$:	TSTB @ML.REG+50		9402
21268	075766	100375			BPL 9\$		
21269	075770	016701	116020		MOV LST.BLK,R1		9405
21270	075774	005201			INC R1		
21271	075776	027701	117432		CMP @ML.REG+30,R1		
21272	076002	001432			BEQ 10\$		
21273	076004	104455			TRAP 55		9408
21274	076006	000145			.WORD 145		
21275	076010	012706			.WORD ASYNC		
21276	076012	026302			.WORD DUMPER		
21277	076014	012746	010664		MOV #WRD.14,-(SP)		9409
21278	076020	012746	011054		MOV #WRD.31,-(SP)		

21342
21343
21347
21348
21352 076254
21353 076254 004767 177060
21354 076260 104466
21355 076262 006000
21356 076264 103773
21357 076266 000207
21358
21359
21360
21365
21366
21367 ; 9434 !<BLF/PAGE>

.SBTTL T43 TEST CODE SECTION

T43::
1\$: JSR PC,\$T43 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

9432

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

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (96)

21369 :ML4AD
21370 :
21371 :
21372 :
21373 :
21374 :
21375 :
21376 :
21377 :
21378 :
21379 :
21380 :
21381 :
21382 :
21383 :
21384 :
21385 :
21386 :
21387 :
21388 :
21389 :
21390 :
21391 :
21392 :
21393 :
21394 :
21395 :
21396 :
21397 :
21398 :
21399 :
21400 :
21401 :
21402 :
21403 :
21404 :
21405 :
21406 :
21407 :
21408 :
21409 :
21410 :
21411 :
21412 :
21413 :
21414 :
21415 :
21416 :
21417 :
21418 :
21419 :
21420 :
21421 :
21422 :
21423 :

9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 44

TEST NAME: INVALID ADRS TEST

TEST DESCRIPTION:

TEST THE DETECTION OF ILLEGAL DSA
ADDRESSES BY:

1. DOING A MBUS WRITE FUNCTION
AT ALL POSSIBLE ILLEGAL DSA
ADDRESSES AND TEST THE
IAE BIT SET.

IMPLICIT INPUTS:

IO_BUF
A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READS AND WRITE
FUNCTION ARE FOUND.

!--

Local

IAE_CNT;

!INVLID ADRS COUNT

CLR_THRESHOLD;

!CLEAR ERROR PRINT THRESHOLD

IAE_CNT = .LST_BLK;

!START AT LAST BLOCK + 1

do

!TEST FOR ALL INVALID ADDRESSES

begin

!INCREMENT IAE_CNT

IAE_CNT = .IAE_CNT + 1;

BGNSUB;

CLR_MBUS;

MLDA = .IAE_CNT;

!LOAD DSA

MLWC = not 255;

!LOAD WORD COUNT

MLBA = IO_BUF;

!LOAD MBUS ADRS

MLCS1 = write;

!DO A WRITE FUNCTION

if .IAE IS_NOT_SET

!SET IF IAE SET

then

begin

!ERROR IF NOT SET

CMR_THRESHOLD;

!COMPARE ERROR PRINT THRESHOLD

ERRDF (102, ASYNC, DUMPER);

PRINTB (FIV_FMT, WRD_30, PHR_1, WRD_11, WRD_30, WRD_10);

end;

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (96)

21425 :ML4AD
21426 :
21427 :
21428 :
21429 :
21430 :
21431 :
21432 :
21433 :
21434 :
21435 :
21436 :
21440 :
21441 :

TEST CODE SECTION

9487
9488
9489
9490
9491
9492
9493
9494
9495

ENDSUB;
end

!VERSION CZMLAD CHANGED TEST TO UNSIGNED TEST

until .IAE_CNT eqlU %0'177777':

!REPEAT UNTIL ALL TESTED

ENDTST:

.SBTTL \$T44 TEST CODE SECTION

\$T44: MOV R1,-(SP) :
CLR P.CNT :
MOV LST.BLK,R1 : * ,IAE.CNT
1\$: INC R1 : IAE.CNT
2\$: TRAP 2 :
BISB #40,@ML.REG+40 :
MOV ML.DUT,R0 :
BIC #177770,R0 :
BICB #7,@ML.REG+40 :
BISB R0,@ML.REG+40 :
MOV R1,@ML.REG+30 : IAE.CNT,*
MOV #-400,@ML.REG+10 :
MOV #10.BUF,@ML.REG+20 :
MOV #61,@ML.REG :
BIT #2000,@ML.REG+60 :
BNE 3\$:
INC P.CNT :
CMP P.CNT,LIMIT :
BGT 4\$:
TRAP 5\$:
.WORD 146 :
.WORD ASYNC :
.WORD DUMPER :
MOV #WRD.10,-(SP) :
MOV #WRD.30,-(SP) :
MOV #WRD.11,-(SP) :
MOV #PHR.1,-(SP) :
MOV #WRD.30,-(SP) :
MOV #FIV.FMT,-(SP) :
MOV #6,-(SP) :
MOV SP,R0 : SP,*
TRAP 14 :
ADD #16,SP :
3\$: TRAP 67 :
ROR R0 :

21445 076270 010146
21446 076272 005067 117100
21447 076276 016701 115512
21448 076302 005201
21449 076304 104402
21450 076306 152777 000040 117130
21451 076314 016700 117512
21452 076320 042700 177770
21453 076324 142777 000007 117112
21454 076332 150077 117106
21455 076336 010177 117072
21456 076342 012777 177400 117044
21457 076350 012777 014022 117046
21458 076356 012777 000061 117020
21459 076364 032777 002000 117072
21460 076372 001034
21461 076374 005267 116776
21462 076400 026767 116772 116772
21463 076406 003031
21464 076410 104455
21465 076412 000146
21466 076414 012706
21467 076416 026302
21468 076420 012746 010630
21469 076424 012746 011046
21470 076430 012746 010640
21471 076434 012746 011610
21472 076440 012746 011046
21473 076444 012746 010304
21474 076450 012746 000006
21475 076454 010600
21476 076456 104414
21477 076460 062706 000016
21478 076464 104467
21479 076466 006000

9433
9465
9468
9472
9473
9475
9476
9477
9478
9480
9482
9484
9485
9482
9486

21481
21482
21483
21484 076470 103705
21485 076472 020127 177777
21486 076476 001301
21487 076500 012601
21488 076502 000207

:ML4AD
:
TEST CODE SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

4\$: BLO 2\$
CMP R1,#-1 ; IAE.CNT,*
BNE 1\$
MOV (SP)+,R1 ;
RTS PC 9433

: Routine Size: 70 words
: Maximum stack depth per invocation: 8 words

21489
21490
21491
21496
21497
21501
21502
21506 076504
21507 076504 004767 177560
21508 076510 104466
21509 076512 006000
21510 076514 103773
21511 076516 000207

.SBTTL T44 TEST CODE SECTION

T44::
1\$: JSR PC,ST44 ;
TRAP 66 9493
ROR R0
BLO 1\$
RTS PC

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

21512
21513
21514
21519
21520
21521 : 9496 !<BLF/PAGE>

21523 :ML4AD
21524 :
21525 :
21526 :
21527 :
21528 :
21529 :
21530 :
21531 :
21532 :
21533 :
21534 :
21535 :
21536 :
21537 :
21538 :
21539 :
21540 :
21541 :
21542 :
21543 :
21544 :
21545 :
21546 :
21547 :
21548 :
21549 :
21550 :
21551 :
21552 :
21553 :
21554 :
21555 :
21556 :
21557 :
21558 :
21559 :
21560 :
21561 :
21565 :
21566 :

TEST CODE SECTION

```

9497 !
9498 BGNTST;
9499
9500 !++
9501 TEST NUMBER: TST 45
9502
9503 TEST NAME: ADRS OVERFLOW BIT TEST
9504
9505 TEST DESCRIPTION:
9506 TEST THE DETECTION OF ADDRESS OVERFLOWS BY:
9507
9508 1. STARTING AT THE LAST BLOCK DO A TWO BLOCK TRANSFER.
9509
9510 2. READ THE AOE BIT SET.
9511
9512 IMPLICIT INPUTS:
9513 IO BUF
9514 A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITE
9515 FUNCTIONS ARE FOUND.
9516
9517
9518 CLR MBUS;
9519 MLWC = not 511;
9520 MLBA = IO BUF;
9521 MLDA = .LST_BLK;
9522 MLCS1 = write;
9523 TIME_OUT_LOOP;
9524
9525 i+ .AOE IS_NOT_SET
9526 then
9527 begin
9528 ERRDF (103, SYNC, DUMPER);
9529 PRINTB (FOR_FMT, WRD_26, PHR_1, WRD_11, FNC_19);
9530 end;
9531
9532 ENDTST;
    
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (97)

```

!DO TWO BLOCK XFERR
!LOAD MBUS ADRS
!START AT LAST BLOCK
!DO A WRITE FUNCTION
    
```

```

!SEE IF AOE SET
!ERROR IF NOT SET
    
```

Address	Hex	Dec	Hex	Hex	Label	Code	Comment
21570	076520	152777	000040	116716	\$T45:	.SBTTL	\$T45 TEST CODE SECTION
21571	076526	016700	117300			BISB	#40,@ML.REG+40
21572	076532	042700	177770			MOV	ML,DUT,RO
21573	076536	142777	000007	116700		BIC	#177770,RO
21574	076544	150077	116674			BICB	#7,@ML.REG+40
21575	076550	012777	177000	116636		BISB	RO,@ML.REG+40
21576	076556	012777	014022	116640		MOV	#-1000,@ML.REG+10
21577	076564	016777	115224	116642		MOV	#IO.BUF,@ML.REG+20
							LST.BLK,@ML.REG+30

9498

9519
9520
9521

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

```

21579      ;ML4AD
21580      ;
21581      ;
21582 076572 012777 000061 116604      1$:  MOV    #61,@ML.REG      ;
21583 076600 105777 116650      TSTB  @ML.REG+50      ;
21584 076604 100375      BPL   1$              ;
21585 076606 032777 001000 116650      BIT   #1000,@ML.REG+60 ;
21586 076614 001024      BNE   2$              ;
21587 076616 104455      TRAP  55              ;
21588 076620 000147      .WORD 147            ;
21589 076622 012750      .WORD SYNC           ;
21590 076624 026302      .WORD DUMPER         ;
21591 076626 012746 012426      MOV   #FNC.19,-(SP)  ;
21592 076632 012746 010640      MOV   #WRD.11,-(SP) ;
21593 076636 012746 011610      MOV   #PHR.1,-(SP)  ;
21594 076642 012746 011016      MOV   #WRD.26,-(SP) ;
21595 076646 012746 010270      MOV   #FOR.FMT,-(SP) ;
21596 076652 012746 000005      MOV   #5,-(SP)      ;
21597 076656 010600      MOV   SP,R0         ; SP,*
21598 076660 104414      TRAP  14             ;
21599 076662 062706 000014      ADD   #14,SP         ;
21600 076666 000207      RTS   PC             ;

```

9522
9525
9528
9529
9527
9495

: Routine Size: 52 words
: Maximum stack depth per invocation: 6 words

.SBTTL T45 TEST CODE SECTION

```

21614
21618 076670
21619 076670 004767 177624      T45:: 1$: JSR   PC,$T45      ;
21620 076674 104466      TRAP  66             ;
21621 076676 006000      ROR   R0              ;
21622 076700 103773      BLO   1$              ;
21623 076702 000207      RTS   PC              ;

```

9530

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

21633 : 9533 !<BLF/PAGE>

21635 :ML4AD

29-Mar-1982 16:23:04

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (98)

29-Mar-1982 16:21:03

TEST CODE SECTION

```

21636 :
21637 :
21638 : 9534 !
21639 : 9535 !BGNTST;
21640 : 9536 !
21641 : 9537 !++
21642 : 9538 !TEST NUMBER: TST 46
21643 : 9539 !
21644 : 9540 !TEST NAME: SYNC BUS PARITY TEST
21645 : 9541 !
21646 : 9542 !TEST DESCRIPTION:
21647 : 9543 !TEST ABILITY OF SYNC BUS TO DETECT AND GENERATE
21648 : 9544 !GOOD PARITY BY:
21649 : 9545 !
21650 : 9546 ! 1. VIA MBUS WRITE FUNCTION WRITE ALTERNATING ONES AND ZEROES TO THE
21651 : 9547 !DEVICE AND READ THE DPAR BIT CLEARED.
21652 : 9548 !
21653 : 9549 ! 2. REPEAT WITH SHIFTED DATA
21654 : 9550 !
21655 : 9551 ! 3. VIA MBUS READ FUNCTION READ THE ALTERNATING PATTERN AND MDPE CLEARED.
21656 : 9552 !
21657 : 9553 !IMPLICIT INPUTS:
21658 : 9554 !IO BUF
21659 : 9555 !A VECTOR OF 256 WORDS WHERE DATA FOR MBUS READS AND WRITES
21660 : 9556 !FUNCTIONS ARE FOUND.
21661 : 9557 !--
21662 : 9558 !
21663 : 9559 !if .PAR_DIS IS_SET !SEE IF PARITY IS DISABLED
21664 : 9560 !then
21665 : 9561 !begin !PRINT MESSAGE AND EXIT TST IF YES
21666 : 9562 !PRINTB (THR_FMT, FNC_3, WRD_7, WRD_37);
21667 : 9563 !EXIT_TST;
21668 : 9564 !end;
21669 : 9565 !
21670 : 9566 !CLR_MBUS;
21671 : 9567 !BAI = ONE; !SET ON FIRST IO BUF ADRS
21672 : 9568 !IO_BUF = %o'125252'; !ALTERNATE 1, 0 PATTERN
21673 : 9569 !
21674 : 9570 !incr TWICE from 0 to 1 do !REPEAT LOOP TWICE
21675 : 9571 !begin
21676 : 9572 !BGNSUB;
21677 : 9573 !GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
21678 : 9574 !MLCS1 = write; !DO A WRITE FUNCTION
21679 : 9575 !TIME_OUT_LOOP;
21680 : 9576 !
21681 : 9577 !if .DPAR IS_SET !SEE IF DPAR GOT SET
21682 : 9578 !then
21683 : 9579 !begin !ERROR IF SET
21684 : 9580 !ERRDF (105, SYNC, DUMPER);
21685 : 9581 !PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_9);
21686 : 9582 !end;
21687 : 9583 !
21688 : 9584 !IO_BUF = .IO_BUF^ONE; !SHIFT THE IO BUF & REPEAT
21689 : 9585 !MLER = ZEROES; !CLEAR OUT ERROR REG & REPEAT

```

```

21691 :ML4AD
21692 :
21693 :
21694 : 9586 ENDSUB;
21695 : 9587 end;
21696 : 9588
21697 : 9589 incr TWICE from 0 to 1 do !REPEATE LOOP TWICE
21698 : 9590 begin
21699 : 9591 BGNSUB;
21700 : 9592 CLR_MBUS;
21701 : 9593 BAI = ONE; !SET ON FIRST IO BUF ADRS
21702 : 9594 GD_BLK_XFER (); !SET UP A GOOD BLOCK XFERR
21703 : 9595 MLCS1 = read; !DO A READ XFERR
21704 : 9596 TIME_OUT_LOOP;
21705 : 9597
21706 : 9598 if .MDPE IS_SET !SEE IF READ GENERATED BAD PARITY
21707 : 9599 then
21708 : 9600 begin !ERROR IF MDPE SET
21709 : 9601 ERRDF (106, SYNC, DUMPER);
21710 : 9602 PRINTB (FOR_FMT, WRD_23, WRD_6, WRD_7, WRD_8);
21711 : 9603 end;
21712 : 9604
21713 : 9605 ENDSUB;
21714 : 9606 IO_BUF = .IO_BUF^ONE; !MAKE DATA PATTERN HAVE ONE LESS ONE
21715 : 9607 GD_BLK_XFER (); !SET UP A GOOD BLK XFER
21716 : 9608 MLCS1 = write; !CHANGE THE BACKGROUND IN MEMORY
21717 : 9609 TIME_OUT_LOOP;
21718 : 9610 end;
21719 : 9611
21720 : 9612 ENDTST;

```

```

21725
21729 076704 010146
21730 076706 026727 115076 000001 $T46:
21731 076714 001021
21732 076716 012746 011122
21733 076722 012746 010566
21734 076726 012746 012172
21735 076732 012746 010256
21736 076736 012746 000004
21737 076742 010600
21738 076744 104414
21739 076746 104463
21740 076750 062706 000012
21741 076754 000167 000402
21742 076760 152777 000040 116456 1$:
21743 076766 016701 117040
21744 076772 042701 177770
21745 076776 142777 000007 116440

.SBTTL $T46 TEST CODE SECTION
MOV R1, -(SP)
CMP PAR.DIS, #1
BNE 1$
MOV #WRD.37, -(SP)
MOV #WRD.7, -(SP)
MOV #FNC.3, -(SP)
MOV #THR.FMT, -(SP)
MOV #4, -(SP)
MOV SP, R0
TRAP 14
TRAP 63
ADD #12, SP
JMP 9$
BISB #40, 2ML.REG+40
MOV ML.DUT, R1
BIC #177770, R1
BICB #7, 2ML.REG+40

```

9532
9559
9562
9559
9561
9564


```

21803      ;ML4AD
21804      ;
21805      ;
21806 077270 012746 010560      MOV      #WRD.6, -(SP)
21807 077274 012746 010766      MOV      #WRD.23, -(SP)
21808 077300 012746 010270      MOV      #FOR.FMT, -(SP)
21809 077304 012746 000005      MOV      #5, -(SP)
21810 077310 010600              MOV      SP, R0
21811 077312 104414              TRAP     14
21812 077314 062706 000014      ADD      #14, SP
21813 077320 104467      7$:     TRAP     67
21814 077322 006000              ROR      R0
21815 077324 103715              BLO     5$
21816 077326 006367 114470      ASL     IO.BUF
21817 077332 004767 120754      JSR     PC, GD.BLK.XFER
21818 077336 012777 000061 116040      MOV     #61, @ML.REG
21819 077344 105777 116104      8$:     TSTB   @ML.REG+50
21820 077350 100375              BPL     8$
21821 077352 005201              INC     R1
21822 077354 020127 000001      CMP     R1, #1
21823 077360 003677              BLE     5$
21824 077362 012601      9$:     MOV     (SP)+, R1
21825 077364 000207              RTS     PC
21826
21827      ; Routine Size: 153 words
21828      ; Maximum stack depth per invocation: 7 words
21833
21834
21838
21839      .SBTTL  T46 TEST CODE SECTION
21843 077366      T46::
21844 077366 004767 177312      1$:     JSR     PC, ST46
21845 077372 104466              TRAP     66
21846 077374 006000              ROR      R0
21847 077376 103773              BLO     1$
21848 077400 000207              RTS     PC
21849
21850      ; Routine Size: 6 words
21851      ; Maximum stack depth per invocation: 0 words
21856
21857 :          9613  !<BLF/PAGE>

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (99)

21859 :ML4AD
21860 :
21861 :
21862 :
21863 :
21864 :
21865 :
21866 :
21867 :
21868 :
21869 :
21870 :
21871 :
21872 :
21873 :
21874 :
21875 :
21876 :
21877 :
21878 :
21879 :
21880 :
21881 :
21882 :
21883 :
21884 :
21885 :
21886 :
21887 :
21888 :
21889 :
21890 :
21891 :
21892 :
21893 :
21894 :
21895 :
21896 :
21897 :
21898 :
21899 :
21900 :
21901 :
21902 :
21903 :
21904 :
21905 :
21906 :
21907 :
21908 :
21909 :
21910 :
21911 :
21912 :
21913 :

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 47

TEST NAME: WRITE READ ML11

TEST DESCRIPTION:

PROVIDES A MBUS READ/WRITE
DATA TRANSFER TROUBLE SHOOTING
LOOP BY:

1. LOAD APPROPRIATE RH REGISTERS.
DO A WRITE FUNCTION.
2. LOAD APPROPRIATE RH REGISTERS
DO A WRITE CHECK FUNCTION.
3. COMPLIMENT DATA AND
REPEAT.

IMPLICIT INPUTS:

IO BUF
A VECTOR OF 256 WORDS WHERE
DATA FOR MBUS READ AND WRITE
FUNCTIONS IS FOUND.

!--

Local

TST_PAT;

!TEST PATTERN

TST_PAT = %o'052525';

incr TWICE from 0 to 1 do

!WRITE READ 1'S AND 0'S ON MBUS

begin

CLR_MBUS;

BAI = ONE;

IO_BUF = TST_PAT;

GD_BLK_XFER ();

MLCST = write;

TIME_OUT_LOOP;

CLR_MBUS;

BAI = ONE;

GD_BLK_XFER ();

MLCST = WRT_CHK;

TIME_OUT_LOOP;

!SET ON FIRST IO BUF ADRS
!LOAD FIRST IO_BUF ADRS
!SET UP A GOOD_BLOCK XFERR
!DO A WRITE FUNCTION

!SET ON FIRST IO BUF ADRS
!SET UP A GOOD_BLOCK XFERR
!DO A WRITE CHECK FUNCTION

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (99)

```

21915 :ML4AD
21916 :
21917 :
21918 : 9666 if .WCE IS_SET !SEE IF WRITE CHECK ERROR SET
21919 : 9667 then
21920 : 9668 begin !ERROR IF SET
21921 : 9669 ERRDF (109, TRBLE LOOP, DUMPFR);
21922 : 9670 PRINTB (SIX_FMT, FNC_4, WRD_10, WRD_12, FNC_5, FNC_6, FNC_3);
21923 : 9671 end;
21924 : 9672
21925 : 9673 TST_PAT = not .TST_PAT; !COMPLIMENT TST_PAT AND REPEAT
21926 : 9674 end;
21927 : 9675
21928 : 9676
21932 :
21933 :

```

```

21937 077402 004167 105132 ST47: .SBTTL $T47 TEST CODE SECTION
21938 077406 012701 052525 JSR R1,$SAVE2
21939 077412 005002 MOV #52525,R1 :
21940 077414 152777 000040 116022 1$: CLR R2 : *TST.PAT
21941 077422 016700 116404 BISB #40,AML.REG+40 : TWICE
21942 077426 042700 177770 MOV ML,DUT,RO :
21943 077432 142777 000007 116004 BIC #177770,RO :
21944 077440 150077 116000 BISB R0,AML.REG+40 :
21945 077444 152777 000010 115772 BISB #10,AML.REG+40 :
21946 077452 010167 114344 MOV R1,IO.BUF : TST.PAT,*
21947 077456 004767 120630 JSR PC,GD.BLK.XFER :
21948 077462 012777 000061 115714 MOV #61,AML.REG :
21949 077470 105777 115760 2$: TSTB AML.REG+50 :
21950 077474 100375 BPL 2$ :
21951 077476 152777 000040 115740 BISB #40,AML.REG+40 :
21952 077504 016700 116322 MOV ML,DUT,RO :
21953 077510 042700 177770 BIC #177770,RO :
21954 077514 142777 000007 115722 BICB #7,AML.REG+40 :
21955 077522 150077 115716 BISB R0,AML.REG+40 :
21956 077526 152777 000010 115710 BISB #10,AML.REG+40 :
21957 077534 004767 120552 JSR PC,GD.BLK.XFER :
21958 077540 012777 000051 115636 MOV #51,AML.REG :
21959 077546 105777 115702 3$: TSTB AML.REG+50 :
21960 077552 100375 BPL 3$ :
21961 077554 032777 040000 115662 BIT #40000,AML.REG+40 :
21962 077562 001430 BEQ 4$ :
21963 077564 104455 TRAP 5$ :
21964 077566 000155 .WORD 155 :
21965 077570 013250 .WORD TRBLE_LOOP :
21966 077572 026792 .WORD DUMPER :
21967 077574 012746 012172 MOV #FNC.3,-(SP) :
21968 077600 012746 012226 MOV #FNC.6,-(SP) :
21969 077604 012746 012216 MOV #FNC.5,-(SP) :

```

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

21971			
21972			
21973			
21974	077610	012746	010650
21975	077614	012746	010630
21976	077620	012746	012200
21977	077624	012746	010322
21978	077630	012746	000007
21979	077634	010600	
21980	077636	104414	
21981	077640	062706	000020
21982	077644	005101	
21983	077646	005202	
21984	077650	020227	000001
21985	077654	003657	
21986	077656	000207	

:ML4AD
: TEST CODE SECTION

```

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
COM R1
INC R2
CMP R2,#1
BLE 1$
RTS PC

```

: SP,*
:
: TST.PAT
: TWICE
: TWICE,*
:

9668
9673
9652

9612

: Routine Size: 87 words
: Maximum stack depth per invocation: 11 words

21987			
21988			
21989			
21994			
21995			
21999			
22000			
22004	077660		
22005	077660	004767	177516
22006	077664	104466	
22007	077666	006000	
22008	077670	103773	
22009	077672	000207	

```

.SBTTL T47 TEST CODE SECTION
T47::
1$: JSR PC,$T47
TRAP 66
ROR R0
BLO 1$
RTS PC

```

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

9674

22010
22011
22012
22017
22018
22019 : 9677 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (100)

22021 :ML4AD
22022 :
22023 :
22024 :
22025 :
22026 :
22027 :
22028 :
22029 :
22030 :
22031 :
22032 :
22033 :
22034 :
22035 :
22036 :
22037 :
22038 :
22039 :
22040 :
22041 :
22042 :
22043 :
22044 :
22045 :
22046 :
22047 :
22048 :
22049 :
22050 :
22051 :
22052 :
22053 :
22054 :
22055 :
22056 :
22057 :
22058 :
22059 :
22060 :
22061 :
22062 :
22063 :
22064 :
22065 :
22066 :
22067 :
22068 :
22069 :
22070 :
22071 :
22072 :
22073 :
22074 :
22075 :

9678
9679
9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9700
9701
9702
9703
9704
9705
9706
9707
9708
9709
9710
9711
9712
9713
9714
9715
9716
9717
9718
9719
9720
9721
9722
9723
9724
9725
9726
9727
9728
9729

TEST CODE SECTION

BGNTST;

!++

TEST NUMBER: TST 48

TEST NAME: CRC DATA BUS TEST

TEST DESCRIPTION:

TEST THE CRC DATA BUS BETWEEN THE CRC GENERATORS AND THE CRC/MBUS DATA MUX FOR CONTINUITY AND BIT UNIQUENESS BY:

1. WHILE IN ECC DIAGNOSTIC MODE DO GOOD BLOCK TRANSFERS WITH VARIOUS CRC A,CRC B AND PAR CRC WRD WHICH WILL EXERCISE BUS CONTINUITY AND UNIQUENESS.
2. THEN IN DATA DIAGNOSTEC MODE READ THE CRC NIBBLES FROM ONE CRC GROUP FOR CORRECT CRC_A,CRC_B AND PAR_CRC_WRD DATA PATTERNS.

IMPLICIT INPUTS:

PD TEMP
A BIT VECTOR OF 16 BITS WHERE THE READ PROM DATA IS STORED OAND ACCESSED FROM.

!--

local

TMP_E1,
DODD_FLG,
CRC_NIB,
CRC_TSTED,
ERR_FLG,
NIB_PAT;

CLR_THRESHOLD;

CRC_NIB = 9;

DODD_FLG = ZERO;

incr LOOP from 0 to 4 do

begin

BGNSUB;

CLR_MBUS;

ECC_DM = ONE;

case .LOOP from 0 to 4 of

!TEMP STORAGE FOR E1 DATA WORD
!DROP UNIT FLAG
!POINTER TO THE CRC NIBBLE
!STORES HOW MANY CRC NIBBLES WE HAVE TESTED
!ERROR FLAG
!TEST DATA FOR THE CRC NIBBLE

!CLEAR ERROR PRINT THRESHOLD
!SET CRC NIBBLE POINTER
!CLEAR THE DROP UNIT FLAG

!REPEAT THE TEST WITH FIVE NIBBLE PATTERNS

!START OF THE SCOPE LOOP

!SET ECC DIAG MOD

!SELECT ONE OF THE FIVE NIBBLE PATTERNS

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (100)

22077 :ML4AD
22078 :
22079 :
22080 :
22081 :
22082 :
22083 :
22084 :
22085 :
22086 :
22087 :
22088 :
22089 :
22090 :
22091 :
22092 :
22093 :
22094 :
22095 :
22096 :
22097 :
22098 :
22099 :
22100 :
22101 :
22102 :
22103 :
22104 :
22105 :
22106 :
22107 :
22108 :
22109 :
22110 :
22111 :
22112 :
22113 :
22114 :
22115 :
22116 :
22117 :
22118 :
22119 :
22120 :
22121 :
22122 :
22123 :
22124 :
22125 :
22126 :
22127 :
22128 :
22129 :
22130 :
22131 :

TEST CODE SECTION

set
[0] :
NIB_PAT = %b'110';

!PATTERN 0

[1] :
begin
MLE1 = ONES;
MLE2 = ONES;
NIB_PAT = %b'001';
end;

!PATTERN 1

[2] :
begin
CRC_A = ONES;
CRC_B = ONES;
NIB_PAT = %b'000';
end;

!PATTERN 2

[3] :
begin
PAR_CRC_WRD = ONES;
CRC_B = ONES;
NIB_PAT = %b'011';
end;

!PATTERN 3

[4] :
begin

!PATTERN 4

!+ THE FOLLOWING ASSIGNMENT:

MLE1 = .TMP_E1

IS EQUIVALENT TO THE FOLLOWING
TWO ASSIGNMENTS:

PAR_CRC_WRD = ONES;
CRC_A = ONES;

!LOAD PAR_CRC_WRD WITH REM_TBL
!LOAD CRC_A WITH REM_TBL

THIS IS NECESSARY DUE TO THE FACT THAT
THE E1 REGISTER IS WORD ORIENTATED AND
THE BLISS COMPILER GENERATES BYTE INST
TO ACCESS THE REGISTER.

TMP_E1 = ZEROES;
TMP_E1<0, 6> = ONES;
TMP_E1<8, 6> = ONES;
MLE1 = .TMP_E1;

!CLEAR OUT THE TEMP WORD
!LOAD CRC_A WITH REM_TBL
!LOAD PAR_CRC_WRD WITH REM_TBL
!LOAD THE E1 REGISTER WITH DATA

NIB_PAT = %b'101';
end;

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (100)

```

22133 :ML4AD
22134 :
22135 :
22136 : 9782
22137 : 9783
22138 : 9784
22139 : 9785
22140 : 9786
22141 : 9787
22142 : 9788
22143 : 9789
22144 : 9790
22145 : 9791
22146 : 9792
22147 : 9793
22148 : 9794
22149 : 9795
22150 : 9796
22151 : 9797
22152 : 9798
22153 : 9799
22154 : 9800
22155 : 9801
22156 : 9802
22157 : 9803
22158 : 9804
22159 : 9805
22160 : 9806
22161 : 9807
22162 : 9808
22163 : 9809
22164 : 9810
22165 : 9811
22166 : 9812
22167 : 9813
22168 : 9814
22169 : 9815
22170 : 9816
22171 : 9817
22172 : 9818
22173 : 9819
22174 : 9820
22175 : 9821
22176 : 9822
22177 : 9823
22178 : 9824
22179 : 9825
22180 : 9826
22181 : 9827
22182 : 9828
22183 : 9829
22184 : 9830
22185 : 9831

```

```

TEST CODE SECTION
tes:
GD_BLK_XFER ();
MLCS1 = write;
TIME_OUT_LOOP;
DAT_DM = ONE;
GD_BLK_XFER ();
MLCS1 = read;
CRC_TSTED = ZERO;
DELAY (ONE_US);

do
begin
PD_TEMP = .MLPD;
DAT_CLK = ONE;
DELAY (ONE_US);
RD_LNG_WRD;

if .PD_TEMP [.CRC_NIB] IS_NOT_SET
then
begin
CRC_TSTED = .CRC_TSTED + 1;
TST_LNG_WRD (.CRC_NIB, .NIB_PAT, ERR_FLG);

if .ERR_FLG IS_SET
then
begin
CMP THRESHOLD;
ERRDF (131, SYNC, DUMPER);
PRINTB (FOR_FMT, WRD 64, WRD 24, WRD 73, WRD 10);
PRINTB (FMT_5, .NIB_PAT, .NIB_SAVE [NIB 9], 8);
DODU_FLG = ONE;
end;
end;

until .CRC_TSTED eql 6;

ENDSUB;
end;

if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN);
DOCLN;
end;

ENDTST;

```

```

!SET UP A GOOD BLOCK TRANSFER
!DO A WRITE TRANSFER
!WAIT UNTIL DRIVE READY
!SET DATA DIAG MODE
!SET UP A GOOD BLOCK TRANSFER
!DO A READ TRANSFER

!TEST ONE CRC GROUP '6 CRC NIBBLES'

!READ THE PROM DATA FOR THIS ARRAY WORD
!CLOCK OUT AN ARRAY WORD INTO THE DATA DIAG REG'S
!DELAY ONE MICRO SECOND
!READ THE DATA DIAG REG'S INTO MEMORY

!TEST THIS CRC NIBBLE IF GOOD

!INCREMENT THE GOOD NIBBLE COUNTER
!TEST THE CRC NIBBLE

!SEE IF THE TEST FOUND AN ERROR

!REPORT ERROR IF FLAG IS SET
!COMPARE ERROR PRINT THRESHOLD

!SET THE DROP UNIT FLAG

!REPEAT UNTIL ONE CRC GROUP HAS BEEN TESTED

!END OF SCOPE LOOP

```

Address	Op1	Op2	Op3	Op4	Label	Instruction	Comment	Address
22193						.SBTTL	ST48 TEST CODE SECTION	
22194								
22198	077674	004167	104712		ST48:	JSR	R1,\$SAVE5	9676
22199	077700	162706	000012			SUB	#12,SP	
22200	077704	005067	115466			CLR	P,CNT	
22201	077710	012746	000011			MOV	#11,-(SP)	9717
22202	077714	005066	000006			CLR	6(SP)	9720
22203	077720	005001				CLR	R1	9721
22204	077722	104402			1S:	TRAP	2	9723
22205	077724	152777	000040	115512		BISB	#40,@ML.REG+40	9724
22206	077732	016705	116074			MOV	ML,DUT,R5	9725
22207	077736	042705	177770			BIC	#177770,R5	
22208	077742	142777	000007	115474		BICB	#7,@ML.REG+40	
22209	077750	150577	115470			BISB	R5,@ML.REG+40	
22210	077754	152777	000001	115542		BISB	#1,@ML.REG+120	
22211	077762	010105				MOV	R1,R5	9727
22212	077764	006305				ASL	R5	9729
22213	077766	066507	077772			ADD	2\$(R5),PC	
22214	077772	000012			2S:	.WORD	3S-2S	
22215	077774	000020				.WORD	4S-2S	
22216	077776	000042				.WORD	5S-2S	
22217	100000	000062				.WORD	6S-2S	
22218	100002	000104				.WORD	7S-2S	
22219	100004	012703	000006		3S:	MOV	#6,R3	9733
22220	100010	000444				BR	8S	9729
22221	100012	012777	177777	115534	4S:	MOV	#-1,@ML.REG+150	9737
22222	100020	012777	177777	115536		MOV	#-1,@ML.REG+160	9738
22223	100026	012703	000001			MOV	#1,R3	9739
22224	100032	000433				BR	8S	9729
22225	100034	152777	J00077	115512	5S:	BISB	#77,@ML.REG+150	9744
22226	100042	152777	000077	115514		BISB	#77,@ML.REG+160	9745
22227	100050	050003				CLR	R3	9746
22228	100052	J00023				BR	8S	9729
22229	100054	052777	037400	115472	6S:	BIS	#37400,@ML.REG+150	9751
22230	100062	152777	000077	115474		BISB	#77,@ML.REG+160	9752
22231	100070	012703	000003			MOV	#3,R3	9753
22232	100074	000412				BR	8S	9729
22233	100076	005066	000002		7S:	CLR	2(SP)	9775
22234	100102	052766	037477	000002		BIS	#37477,2(SP)	9777
22235	100110	016677	000002	115436		MOV	2(SP),@ML.REG+150	9778
22236	100116	012703	000005			MOV	#5,R3	9780
22237	100122	004767	120164		8S:	JSR	PC,GD.BLK.XFER	9784
22238	100126	012777	000061	115250		MOV	#61,@ML.REG	9785
22239	100134	105777	115314		9S:	TSTB	@ML.REG+50	
22240	100140	100375				BPL	9S	
22241	100142	152777	000010	115354		BISB	#10,@ML.REG+120	9787
22242	100150	004767	120136			JSR	PC,GD.BLK.XFER	9788
22243	100154	012777	000071	115222		MOV	#71,@ML.REG	9789

22362
22363
22367 100640
22368 100640 004767 177030
22369 100644 104466
22370 100646 006000
22371 100650 103773
22372 100652 000207
22373
22374
22375
22380
22381
22382 :

.SBTTL T48 TEST CODE SECTION
T48::
1S: JSR PC,\$T48
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

9829

9832 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (101)

22384 :ML4AD
22385 :
22386 :
22387 :
22388 :
22389 :
22390 :
22391 :
22392 :
22393 :
22394 :
22395 :
22396 :
22397 :
22398 :
22399 :
22400 :
22401 :
22402 :
22403 :
22404 :
22405 :
22406 :
22407 :
22408 :
22409 :
22410 :
22411 :
22412 :
22413 :
22414 :
22415 :
22416 :
22417 :
22418 :
22419 :
22420 :
22421 :
22422 :
22423 :
22424 :
22425 :
22426 :
22427 :
22428 :
22429 :
22430 :
22431 :
22432 :
22433 :
22434 :
22435 :
22436 :
22437 :
22438 :

9833 !
9834 !
9835 !
9836 !
9837 !
9838 !
9839 !
9840 !
9841 !
9842 !
9843 !
9844 !
9845 !
9846 !
9847 !
9848 !
9849 !
9850 !
9851 !
9852 !
9853 !
9854 !
9855 !
9856 !
9857 !
9858 !
9859 !
9860 !
9861 !
9862 !
9863 !
9864 !
9865 !
9866 !
9867 !
9868 !
9869 !
9870 !
9871 !
9872 !
9873 !
9874 !
9875 !
9876 !
9877 !
9878 !
9879 !
9880 !
9881 !
9882 !
9883 !
9884 !

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 49

TEST NAME: UNIQUE NIBBLE CRC GENERATION TEST

TEST DESCRIPTION:

TEST CRC CODES GENERATED FOR ONE CRC
GROUP (52 UNIQUE UNIBUS NIBBLES) BY:

1. DOING MASS BUS WRITE TRANSFERSS WITH THE
FIRST IO BUF CRC GROUP HAVING 51 NIBBLES
OF %B'0000' PATTERN AND THE REMAINING NIBBLE
WITH PATTERNS OF %B'0000' TO %B'1111'.
2. THEN IN DATA DIAGNOSTIC MODE CALCULATE
THE CRC CODE FOR THE FIRST CRC
GROUP AND COMPARE IT AGAINST THE
HARDWARE GENERATED CRC CODE.
3. REPEAT PATTERN OF %B'0000' TO %B'1111'
FOR EACH OF THE 52 NIBBLES.

IMPLICIT INPUTS:

IO_BUF
A VECTOR OF 256 WORDS WHERE DATA
FOR MBUS READS AND WRITES TRANSFERS
IS FOUND.

!--

local

DODU_FLG,
TEMP,
NIB_SEL;

!DROP UNIT FLAG
!TEMPORARY STORAGE LOCATION
!SELECTS 4 BITS 'ONE NIBBLE' FROM THE IO_BUF

DODU_FLG = ZEROES;
CLR_THRESHOLD;

!CLEAR THE DROP UNIT FLAG
!CLEAR ERROR PRINT THRESHOLD

incr CNT from 0 to 13 do
IO_BUF [.CNi.] = ZEROES;

!CLEAR THE FIRST 14 IO_BUF WORDS

incr WRD_CNT from 0 to 12 do

!DO THIS TEST ON 13 IO_BUF WORDS

begin
NIB_SEL = -4;

!RESET THE NIBBLE SELECTOR

incr NIB_TST from 0 to 3 do
begin

!DO THIS TEST ON 4 NIBBLES PER WORD


```

22440 :ML4AD
22441 :
22442 :
22443 : 9885
22444 : 9886
22445 : 9887
22446 : 9888
22447 : 9889
22448 : 9890
22449 : 9891
22450 : 9892
22451 : 9893
22452 : 9894
22453 : 9895
22454 : 9896
22455 : 9897
22456 : 9898
22457 : 9899
22458 : 9900
22459 : 9901
22460 : 9902
22461 : 9903
22462 : 9904
22463 : 9905
22464 : 9906
22465 : 9907
22466 : 9908
22467 : 9909
22468 : 9910
22469 : 9911
22470 : 9912
22471 : 9913
22472 : 9914
22473 : 9915
22474 : 9916
22475 : 9917
22476 : 9918
22477 : 9919
22478 : 9920
22479 : 9921
22480 : 9922
22481 : 9923
22482 : 9924
22483 : 9925
22484 : 9926
22485 : 9927
22486 : 9928
22487 : 9929
22488 : 9930
22489 : 9931
22490 : 9932
22491 : 9933

TEST CODE SECTION

if (.WRD_CNT eql 12) and (.NIB_TST eql 2) then exitloop; !THE LAST 2 NIBBLES ARE NOT TESTED

NIB_SEL = .NIB_SEL + 4; !INCREMENT THE NIBBLE SELECTED
TEMP = ZEROES; !CLEAR TEMP LOCATION

incr NIB_PAT from %b'0000' to %b'1111' by %b'0001' do !RUN PATTERNS ON THIS NIBBLE
begin
BGNSUB; !START OF SCOPE LOOP
TEMP<.NIB_SEL, 4> = .NIB_PAT; !LOAD PATTERN INTO TEMP AT THIS NIBBLE
IO_BUF [.WRD_CNT] = .TEMP; !LOAD THE GENERATED PATTERN INTO THE IO_BUF
CLR MBUS;
GD_BLK_XFER (); !SET UP A GOOD BLOCK TRANSFER
MLCS1 = write; !DO A WRITE TRANSFER
TIME_OUT_LOOP; !WAIT FOR THE TRANSFER TO COMPLETE
A_GEN = ZEROES; !CLEAR THE GEN & CAL STORAGE LOCATIONS
B_GEN = ZEROES;
P_GEN = ZEROES;
A_CAL = ZEROES;
B_CAL = ZEROES;
P_CAL = ZEROES;
CAL_CRC (); !CALL ROUTINE TO CALCULATE THE CRC CODE FOR THIS PATTERN

if ERR_CHK_CRC ()
then
begin
CMP_THRESHOLD; !NOW SEE IF THE GEN CRC IS SAME AS CAL CRC
ERRDF (132, SYNC, DUMPER); !REPORT THE ERROR IF NOT THE SAME
PRINTB (THR_FMT, WRD 5, WRD 64, WRD 8); !COMPARE ERROR PRINT THRESHOLD
PRINTB (FMT_19, .B_GEN, .A_GEN, .P_GEN);
PRINTB (FMT_20, .B_CAL, .A_CAL, .P_CAL);
DODU_FLG = ONE;
end;

ENDSUB; !END OF SCOPE LOOP
end;

IO_BUF [.WRD_CNT] = ZEROES; !CLEAR THIS IO_BUF WORD FO NEXT NIBBLE
end;

end;

if .DODU_FLG IS_SET
then
begin
DODU (.ML_LUN); !DROP THIS UNIT IF THE DODU FLAG GOT SET
DOCLN;
end;

ENDTST;

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (101)

Address	OpCode	Op2	Op3	Op4	Label	Instruction	Comments	Seq
22496								
22497								
22498								
22499								
22500								
22504	100654	004167	103732		ST49:	.SBTTL \$T49 TEST CODE SECTION		
22505	100660	024646				JSR R1,\$SAVE5	:	9831
22506	100662	005066	000002			CMP -(SP),-(SP)	:	
22507	100666	005067	114504			CLR 2(SP)	: DODU.FLG	9872
22508	100672	005000				CLR P.CNT	:	
22509	100674	010001			1\$:	CLR R0	: CNT	9875
22510	100676	006301				MOV R0,R1	: CNT,*	9876
22511	100700	005061	014022			ASL R1	:	
22512	100704	005200				CLR IO.BUF(R1)	:	
22513	100706	020027	000015			INC R0	: CNT	9875
22514	100712	003770				CMP R0,#15	: CNT,*	
22515	100714	005002				BLE 1\$:	
22516	100716	012716	177774		2\$:	CLR R2	: WRD.CNT	9878
22517	100722	010201				MOV #-4,(SP)	: *,NIB.SEL	9880
22518	100724	006301				MOV R2,R1	: WRD.CNT,*	9894
22519	100726	005004				ASL R1	:	
22520	100730	020227	000014		3\$:	CLR R4	: NIB.TST	9882
22521	100734	001003				CMP R2,#14	: WRD.CNT,*	9885
22522	100736	020427	000002			BNE 4\$:	
22523	100742	001575				CMP R4,#2	: NIB.TST,*	
22524	100744	062716	000004		4\$:	BEQ 10\$:	
22525	100750	005005				ADD #4,(SP)	: *,NIB.SEL	9887
22526	100752	005003				CLR R5	: TEMP	9888
22527	100754	104402			5\$:	CLR R3	: NIB.PAT	9890
22528	100756	010546				TRAP 2	:	9891
22529	100760	016646	000002			MOV R5,-(SP)	: TEMP,*	9893
22530	100764	012746	000004			MOV 2(SP),-(SP)	: NIB.SEL,*	
22531	100770	010346				MOV #4,-(SP)	:	
22532	100772	004767	103000			MOV R3,-(SP)	: NIB.PAT,*	
22533	100776	010005				JSR PC,BL\$PU1	:	
22534	101000	010561	014022			MOV R0,R5	: *,TEMP	9891
22535	101004	152777	000040	114432		MOV R5,IO.BUF(R1)	: TEMP,*	9894
22536	101012	016700	115014			BISB #40,@ML.REG+40	:	
22537	101016	042700	177770			MOV ML.DUT,R0	:	
22538	101022	142777	000007	114414		BIC #177770,R0	:	
22539	101030	150077	114410			BICB #7,@ML.REG+40	:	
22540	101034	004767	117252			BISB R0,@ML.REG+40	:	
22541	101040	012777	000061	114336		JSR PC,GD.BLK.XFER	:	9896
22542	101046	105777	114402		6\$:	MOV #61,@ML.REG	:	9897
22543	101052	100375				TSTB @ML.REG+50	:	
22544	101054	005067	114310			BPL 6\$:	
22545	101060	005067	114306			CLR A.GEN	:	9899
22546	101064	005067	114304			CLR B.GEN	:	9900
22547	101070	005067	114266			CLR P.GEN	:	9901
22548	101074	005067	114264			CLR A.CAL	:	9902
22549	101100	005067	114262			CLR B.CAL	:	9903
22550	101104	004767	115732			CLR P.CAL	:	9904
						JSR PC,CAL.CRC	:	9905

22608
22609
22610
22611 101360 001004
22612 101362 016700 114442
22613 101366 104451
22614 101370 104444
22615 101372 022626
22616 101374 000207

;ML4AD
;
TEST CODE SECTION
BNE 12\$
MOV ML.LUN,RO
TRAP 51
TRAP 44
12\$: CMP (SP)+,(SP)+
RTS PC

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

9929
9831

; Routine Size: 169 words
; Maximum stack depth per invocation: 25 words

22617
22618
22619
22624
22625
22629
22630
22634 101376
22635 101376 004767 177252
22636 101402 104466
22637 101404 006000
22638 101406 103773
22639 101410 000207

.SBTTL T49 TEST CODE SECTION
T49::
1\$: JSR PC,\$T49
TRAP 66
ROR R0
BLO 1\$
RTS PC

9931

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

22640
22641
22642
22647
22648
22649 : 9934 !
22650 : 9935 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (102)

22652 :ML4AD
22653 :
22654 :
22655 :
22656 :
22657 :
22658 :
22659 :
22660 :
22661 :
22662 :
22663 :
22664 :
22665 :
22666 :
22667 :
22668 :
22669 :
22670 :
22671 :
22672 :
22673 :
22674 :
22675 :
22676 :
22677 :
22678 :
22679 :
22680 :
22681 :
22682 :
22683 :
22684 :
22685 :
22686 :
22687 :
22688 :
22689 :
22690 :
22691 :
22692 :
22693 :
22694 :
22695 :
22696 :
22697 :
22698 :
22699 :
22700 :
22701 :
22702 :
22703 :
22704 :
22705 :
22706 :

9936
9937
9938
9939
9940
9941
9942
9943
9944
9945
9946
9947
9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987

TEST CODE SECTION

BGNTST:

!++

TEST NUMBER: TST 50

TEST NAME: UNIQUE WORD CRC GENERATION TEST

TEST DESCRIPTION:

TEST CRC CODES GENERATED FOR ONE
CRC GROUP (13 UNIBUS WORDS) BY:

1. DOING MASS BUS WRITE TRANSFERS
USING VARIOUS WORD DATA PATTERNES.
2. THEN IN DATA DIAGNOSTIC MODE
CALCULATE THE CRC CODE FOR THE
FIRST CRC GROUP AND COMPARE IT TO
THE HARDWARE GENERATED CRC CODE.

IMPLICIT INPUTS:

IO_BUF
A VECTOR OF 256 WORDS WHERE DATA
FOR MBUS READS AND WRITES TRANSFERS
IS FOUND.

!--

Local

DODU_FLG,
TST_PAT;

DODU_FLG = ZEROES;

incr PAT_SEL from 0 to 3 do

begin

BGNSUB;

CLR_MBUS;

case .PAT_SEL from 0 to 3 of
set

[0] :
TST_PAT = ONES;

[1] :
TST_PAT = %0'052525';

[2] :
TST_PAT = %0'125252';

[3] :

!DROP UNIT FLAG
!TEST PATTERN SAVE LOCATION

!CLEAR THE DROP UNIT FLAG

!RUN THE TEST ON 4 PATTERNS

!START OF SCOPE LOOP

!SELECT ONE OF THE 4 PATTERNS

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (102)

22708 :ML4AD

TEST CODE SECTION

```

22709 :
22710 :
22711 : 9988          TST_PAT = %o'070707';
22712 : 9989          tes;
22713 : 9990
22714 : 9991          incr CNT from 0 to 255 do
22715 : 9992          IO_BUF [.CNT] = .TST_PAT;
22716 : 9993
22717 : 9994          GD_BLK_XFER ();
22718 : 9995          MLCS1 = write;
22719 : 9996          TIME_OUT_LOOP;
22720 : 9997          A_GEN = ZEROES;
22721 : 9998          B_GEN = ZEROES;
22722 : 9999          P_GEN = ZEROES;
22723 : 10000         A_CAL = ZEROES;
22724 : 10001         B_CAL = ZEROES;
22725 : 10002         P_CAL = ZEROES;
22726 : 10003         CAL_CRC ();
22727 : 10004
22728 : 10005         if ERR_CHK_CRC ()
22729 : 10006         then
22730 : 10007         begin
22731 : 10008         ERRDF (133, SYNC, DUMPER);
22732 : 10009         PRINTB (THR_FMT, WRD 5, WRD 64, WRD 8);
22733 : 10010         PRINTB (FMT_19, .B_GEN, .A_GEN, .P_GEN);
22734 : 10011         PRINTB (FMT_20, .B_CAL, .A_CAL, .P_CAL);
22735 : 10012         DODU_FLG = ONE;
22736 : 10013         end;
22737 : 10014
22738 : 10015         ENDSUB;
22739 : 10016         end;
22740 : 10017
22741 : 10018         if .DODU_FLG IS_SET
22742 : 10019         then
22743 : 10020         begin
22744 : 10021         DODU (.ML_LUN);
22745 : 10022         DOCLN;
22746 : 10023         end;
22747 : 10024
22748 : 10025         ENDTST;
22752 :
22753 :

```

!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

```

22757 101412 004167 103174      ST50:  .SBTTL  ST50 TEST CODE SECTION
22758 101416 005005              JSR    R1,SSAVES
22759 101420 005004              CLR    R5
22760 101422 104402              CLR    R4
22761 101424 152777 000040 114012  ST5:   TRAP  2
22762 101432 016703 114374      BISB  #40,AML.REG+40
                                   MOV    ML.DUT,R3

```

```

:
: DODU.FLG
: PAT.SEL
:

```

9933
996R
9970
9971
9972

22820				:ML4AD				
22821				:	TEST CODE SECTION			
22822								
22823	101704	012746	007632		MOV #FMT.19, -(SP)			
22824	101710	012746	000004		MOV #4, -(SP)			
22825	101714	010600			MOV SP, R0	: SP, *		
22826	101716	104414			TRAP 14	:		
22827	101720	016716	113442		MOV P.CAL, (SP)	:		1001
22828	101724	016746	113432		MOV A.CAL, -(SP)			
22829	101730	016746	113430		MOV B.CAL, -(SP)			
22830	101734	012746	007702		MOV #FMT.20, -(SP)			
22831	101740	012746	000004		MOV #4, -(SP)			
22832	101744	010600			MOV SP, R0	: SP, *		
22833	101746	104414			TRAP 14	:		
22834	101750	012705	000001		MOV #1, R5	: *, DODU.FLG		1001
22835	101754	062706	000032		ADD #32, SP	:		1000
22836	101760	104467		10\$:	TRAP 67	:		1001
22837	101762	006000			ROR R0			
22838	101764	103616			BLO 1\$			
22839	101766	005204			INC R4	: PAT.SEL		9970
22840	101770	020427	000003		CMP R4, #3	: PAT.SEL, *		
22841	101774	003612			BLE 1\$			
22842	101776	005305			DEC R5	: DODU.FLG		1001
22843	102000	001004			BNE 11\$			
22844	102002	016700	114022		MOV ML.LUN, R0	:		1002
22845	102006	104451			TRAP 51			
22846	102010	104444			TRAP 44			
22847	102012	000207		11\$:	RTS PC	:		9933
22848								
22849					: Routine Size: 129 words			
22850					: Maximum stack depth per invocation: 19 words			
22855								
22856								
22860								
22861					.SBTTL T50 TEST CODE SECTION			
22865	102014			T50::				
22866	102014	004767	177372	1\$:	JSR PC, \$T50	:		1002
22867	102020	104466			TRAP 66			
22868	102022	006000			ROR R0			
22869	102024	103773			BLO 1\$			
22870	102026	000207			RTS PC			
22871								
22872					: Routine Size: 6 words			
22873					: Maximum stack depth per invocation: 0 words			

22882
22883
22884 :

10026 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (103)

```

22886 ;ML4AD
22887 ;
22888 ;
22889 : 10027 !
22890 : 10028 BGNST;
22891 : 10029
22892 : 10030 !++
22893 : 10031 TEST NUMBER: TST 51
22894 : 10032
22895 : 10033 TEST NAME: CORRECTABLE ERROR SYNDROME DECODE TEST
22896 : 10034
22897 : 10035 TEST DESCRIPTION:
22898 : 10036 TEST SYNDROME DECODE AND ERROR
22899 : 10037 CORRECTION TO DECODE AND CORRECT
22900 : 10038 SINGLE BIT AND MULTIPLE BIT
22901 : 10039 CHANNEL ERRORS BY:
22902 : 10040
22903 : 10041 1. VIA ECC DIAG MODE FORCE SYNDROME
22904 : 10042 BITS TO DECODE SINGLE BIT AND
22905 : 10043 MULTIPLE BIT CHANNEL ERRORS.
22906 : 10044
22907 : 10045 2. THEN DO A MASS BUS READ TRANSFER
22908 : 10046 AND EXAM THE IO BUF FOR CORRECT
22909 : 10047 BIT COMPLIMENTING.
22910 : 10048
22911 : 10049 IMPLICIT INPUTS:
22912 : 10050 IO_BUF
22913 : 10051 A VECTOR OF 256 WOPDS WHERE DATA
22914 : 10052 FOR MBUS READS AND WRITES TRANSFERS
22915 : 10053 IS FOUND.
22916 : 10054
22917 : 10055
22918 : 10056 !--
22919 : 10057
22920 : 10058 local
22921 : 10059 BITS_XFERED; !POINTS TO THE CHANNEL BEING TESTED
22922 : 10060
22923 : 10061 CLR_THRESHOLD; !CLEAR ERROR PRINT THRESHOLD
22924 : 10062 BAI = ONE; !BEFORE WE START LETS CLEAR OUT THE ML11'S
22925 : 10063 IO_BUF = ZEROES; !GOOD BLOCK BY WRITING ZEROES TO IT
22926 : 10064 GD_BLK_XFER ();
22927 : 10065 MLCS1 = write;
22928 : 10066 TIME_OUT_LOOP;
22929 : 10067 BITS_XFERED = -1; !RESET THE CHANNEL POINTER
22930 : 10068
22931 : 10069 incr PLOG from 0 to 5 do !TEST ONE CRC GROUP '6 ARRAY WORDS'
22932 : 10070 begin
22933 : 10071
22934 : 10072 incr CHANNEL from 0 to 35 do !TEST 36 BITS IN EACH ARRAY WORD
22935 : 10073 begin
22936 : 10074 BGNSUB; !START OF SCOPE LOOP
22937 : 10075 CLR MBUS; !CLEAR THE DRIVE
22938 : 10076 BITS_XFERED = .BITS_XFERED + 1; !INCREMENT THE CHANNEL POINTER
22939 : 10077
22940 : 10078 incr CNT from 0 to 255 do !CLEAR THE FIRST CRC GROUP IN THE IO_BUF

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (103)

```

22942 :ML4AD
22943 :
22944 :
22945 : 10079          IO_BUF [.CNT] = ZEROES;
22946 : 10080
22947 : 10081          FORCE REM (.PLOG, .CHANNEL);
22948 : 10082          GD BLR_XFER ();
22949 : 10083          ECC_DIS = ZERO;
22950 : 10084          MLC51 = read;
22951 : 10085          TIME_OUT_LOOP;
22952 : 10086
22953 : 10087          if not FIND_COMP_BIT (.BITS_XFERED)
22954 : 10088          then
22955 : 10089              begin
22956 : 10090                  CMP THRESHOLD;
22957 : 10091                  ERRDF (134, SYNC, DUMPER);
22958 : 10092                  PRINTB (THR_FMT, WRD 74, WRD 75, PHR_1);
22959 : 10093                  PRINTB (FMT_21, .PLOG, .CHANNEL);
22960 : 10094              end;
22961 : 10095
22962 : 10096          incr WRD_CNT from 0 to 12 do
22963 : 10097
22964 : 10098              if .IO_BUF [.WRD_CNT] neq ZEROES
22965 : 10099              then
22966 : 10100                  begin
22967 : 10101                      CMP THRESHOLD;
22968 : 10102                      ERRDF (135, SYNC, DUMPER);
22969 : 10103                      PRINTB (FOR_FMT, WRD 61, WRD 74, WRD 75, PHR_5);
22970 : 10104                      PRINTB (FMT_21, .PLOG, .CHANNEL);
22971 : 10105                  end;
22972 : 10106
22973 : 10107          ENDSUB;
22974 : 10108          end;
22975 : 10109
22976 : 10110          end;
22977 : 10111
22978 : 10112          BITS_XFERED = -1;
22979 : 10113
22980 : 10114          incr CHANNEL from 0 to 35 do
22981 : 10115              begin
22982 : 10116                  BGNSUB;
22983 : 10117                  CLR MBUS;
22984 : 10118                  BITS_XFERED = .BITS_XFERED + 1;
22985 : 10119
22986 : 10120                  incr CNT from 0 to 20 do
22987 : 10121                      IO_BUF [.CNT] = ZEROES;
22988 : 10122
22989 : 10123                  FORCE REM (58, .CHANNEL);
22990 : 10124                  GD BLR_XFER ();
22991 : 10125                  ECC_DIS = ZERO;
22992 : 10126                  MLC51 = read;
22993 : 10127                  TIME_OUT_LOOP;
22994 : 10128
22995 : 10129                  incr BIT_OFFSET from 0 to 180 by 36 do
22996 : 10130
  
```

```

!FORCE THIS BIT IN THIS WORD IN ERROR
!SET UP A GOOD BLOCK TRANSFER
!CLEAR ECC DISABLE
!DO A READ TRANSFER
!WAIT FOR THE TRANSFER TO COMPLETE

!SEARCH THE IO_BUF FOR THE COMP BIT

!REPORT AN ERROR IF THIS BIT IS NOT COMP
!COMPARE ERROR PRINT THRESHOLD

!SEE IF ANY OTHER BITS GOT COMP'ED

!TEST THE IO_BUF FOR ZEROES

!REPORT AN ERROR IF ANY OTHERS ARE SET
!COMPARE ERROR PRINT THRESHOLD

!END OF SCOPE LOOP

!RESET THE CHANNEL POINTER

!TEST 36 CHANNELS FOR MULTIPLE BIT ERRORS

!START OF SCOPE LOOP
!CLEAR THE DRIVE
!INCREMENT THE CHANNEL POINTER

!CLEAR THE FIRST CRC GROUP IN THE IO_BUF

!FORCE ALL 6 BITS IN THIS CHANNEL IN ERROR
!SET UP A GOOD BLOCK TRANSFER
!CLEAR ECC DISABLE
!DO A READ TRANSFER
!WAIT FOR THE TRANSFER TO CMOplete

!SEE IF EVERY 36TH BIT GOT COMP'ED
  
```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (103)

```

22998 ;ML4AD
22999 :
23000 :
23001 : 10131 if not FIND_COMP_BIT (.BITS_XFERED + .BIT_OFFSET) !SEARCH THE IO_BUF FOR THE COMP'ED BITS
23002 : 10132 then
23003 : 10133 begin !REPORT AN ERROR IF ANY OF THE BITS ARE NOT COMP'ED
23004 : 10134 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23005 : 10135 ERDF (134, SYNC, DUMPER);
23006 : 10136 PRINTB (THR_FMT, WRD_74, WRD_75, PHR_1);
23007 : 10137 PRINTB (FMT_21, 58, .CHANNEL);
23008 : 10138 end;
23009 : 10139
23010 : 10140 incr WRD_CNT from 0 to 12 do !ALSO SEE IF ANY OTHER BIT GOT COMP'ED
23011 : 10141
23012 : 10142 if .IO_BUF [.WRD_CNT] neq ZEROES !TEST THE IO_BUF FOR ALL ZEROES
23013 : 10143 then
23014 : 10144 begin !REPORT AN ERROR IF ANY OTHERS ARE COMP'ED
23015 : 10145 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23016 : 10146 ERDF (135, SYNC, DUMPER);
23017 : 10147 PRINTB (FIV_FMT, WRD_61, WRD_74, WRD_75, WRD_12, WRD_67);
23018 : 10148 PRINTB (FMT_21, 58, .CHANNEL);
23019 : 10149 end;
23020 : 10150
23021 : 10151 ENDSUB; !END OF SCOPE LOOP
23022 : 10152 end;
23023 : 10153
23024 : 10154 ENDTST;
  
```

```

23028
23029
23033 102030 004167 102556 $T51: .SBTTL $T51 TEST CODE SECTION
23034 102034 005746 JSR R1,$SAVE5 : 1002
23035 102036 005067 113334 TST -(SP) :
23036 102042 152777 000010 113374 CLR P.CNT : 1005
23037 102050 005067 111746 BISB #10,@ML.REG+40 : 1006
23038 102054 004767 116232 CLR IO.BUF : 1006
23039 102060 012777 000061 113316 JSR PC,GD,BLK.XFER : 1006
23040 102066 105777 113362 1$: MOV #61,@ML.REG : 1006
23041 102072 100375 TSTB @ML.REG+50
23042 102074 012716 177777 BPL 1$
23043 102100 005005 MOV #-1,(SP) : *.BITS.XFERED 1006
23044 102102 005004 2$: CLR R5 : PLOG 1006
23045 102104 104402 3$: CLR R4 : CHANNEL 1007
23046 102106 152777 000040 113330 TRAP 2 : 1007
23047 102114 016703 113712 BISB #40,@ML.REG+40 : 1007
23048 102120 042703 177770 MOV ML.DUT,R3
23049 102124 142777 000007 113312 BIC #177770,R3
23050 102132 150377 113306 BICB #7,@ML.REG+40
23051 102136 005216 BISB R3,@ML.REG+40
23052 102140 005001 INC (SP) : BITS.XFERED 1007
CLR R1 : CNT 1007
  
```


Address	Hex	Hex	Hex	Label	Code	Comment	Address
23110				:ML4AD			
23111				:	TEST CODE SECTION		
23112							
23113	102376	012750			.WORD SYNC		
23114	102400	026302			.WORD DUMPER		
23115	102402	012746	011714		MOV #PHR.5, -(SP)	:	1010
23116	102406	012746	011552		MOV #WRD.75, -(SP)	:	
23117	102412	012746	011540		MOV #WRD.74, -(SP)	:	
23118	102416	012746	011416		MOV #WRD.61, -(SP)	:	
23119	102422	012746	01027C		MOV #FOR.FMT, -(SP)	:	
23120	102426	012746	000005		MOV #5, -(SP)	:	
23121	102432	010600			MOV SP, R0	: SP, *	
23122	102434	104414			TRAP 14	:	
23123	102436	010416			MOV R4, (SP)	: CHANNEL, *	1010
23124	102440	010546			MOV R5, -(SP)	: PLOG, *	
23125	102442	012746	007754		MOV #FMT.21, -(SP)	:	
23126	102446	012746	000003		MOV #3, -(SP)	:	
23127	102452	010600			MOV SP, R0	: SP, *	
23128	102454	104414			TRAP 14	:	
23129	102456	062706	000022		ADD #22, SP	:	1010
23130	102462	005203		9\$:	INC R3	: WRD.CNT	1009
23131	102464	020327	000014		CMP R3, #14	: WRD.CNT, *	
23132	102470	003725			BLE 8\$:	
23133	102472	022626		10\$:	CMP (SP)+, (SP)+	:	1007
23134	102474	104467			TRAP 67	:	1010
23135	102476	006000			ROR R0	:	
23136	102500	103601			BLO 3\$:	
23137	102502	005204		11\$:	INC R4	: CHANNEL	1007
23138	102504	020427	000043		CMP R4, #43	: CHANNEL, *	
23139	102510	003002			BGT 12\$:	
23140	102512	000167	177366		JMP 3\$:	
23141	102516	005205		12\$:	INC R5	: PLOG	1006
23142	102520	020527	000005		CMP R5, #5	: PLOG, *	
23143	102524	003002			BGT 13\$:	
23144	102526	000167	177350		JMP 2\$:	
23145	102532	012716	177777	13\$:	MOV #-1, (SP)	: *,BITS.XFERED	1011
23146	102536	005003			CLR R3	: CHANNEL	1011
23147	102540	104402		14\$:	TRAP 2	:	1011
23148	102542	152777	000040 112674		BISB #40, @ML.REG+40	:	1011
23149	102550	016705	113256		MOV ML.DUT, R5	:	
23150	102554	042705	177770		BIC #177770, R5	:	
23151	102560	142777	000007 112656		BICB #7, @ML.REG+40	:	
23152	102566	150577	112652		BISB R5, @ML.REG+40	:	
23153	102572	005216			INC (SP)	: BITS.XFERED	1011
23154	102574	005001			CLR R1	: CNT	1012
23155	102576	010102		15\$:	MOV R1, R2	: CNT, *	1012
23156	102600	006302			ASL R2	:	
23157	102602	005062	014022		CLR IO.BUF(R2)	:	
23158	102606	005201			INC R1	: CNT	1012
23159	102610	020127	000024		CMP R1, #24	: CNT, *	
23160	102614	003770			BLE 15\$:	
23161	102616	012746	000072		MOV #72, -(SP)	:	1012
23162	102622	010346			MOV R3, -(SP)	: CHANNEL, *	
23163	102624	004767	115202		JSR PC, FORCE.REM	:	
23164	102630	004767	115456		JSR PC, GD.BLK.XFER	:	1012

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

Address	Instruction	Comments	Address	Instruction	Comments	Address	Instruction	Comments	Date/Time	Page
23166									29-Mar-1982 16:23:04	TOPS
23167									29-Mar-1982 16:21:03	PA:<
23168										
23169	102634	142777	000002	112662	BICB	#2,@ML.REG+120	:			1012
23170	102642	012777	000071	112534	MOV	#71,@ML.REG	:			1012
23171	102650	105777	112600		TSTB	@ML.REG+50				
23172	102654	100375			BPL	16\$				
23173	102656	005002			CLR	R2				
23174	102660	010246			MOV	R2,-(SP)	:	BIT.OFFSET		1012
23175	102662	066616	000006		ADD	6(SP),(SP)	:	BIT.OFFSET,*		1013
23176	102666	004767	114766		JSR	PC,FIND.COMP.BIT		BITS.XFERED,*		
23177	102672	005726			TST	(SP)+				
23178	102674	006000			ROR	R0				
23179	102676	103441			BLO	18\$				
23180	102700	005267	112472		INC	P.CNT	:			1013
23181	102704	026767	112466	112466	CMP	P.CNT,LIMIT				
23182	102712	003040			BGT	19\$				
23183	102714	104455			TRAP	55	:			1013
23184	102716	000206			.WORD	206				
23185	102720	012750			.WORD	SYNC				
23186	102722	026302			.WORD	DUMPER				
23187	102724	012746	011610		MOV	#PHR.1-(SP)	:			1013
23188	102730	012746	011552		MOV	#WRD.75,-(SP)				
23189	102734	012746	011540		MOV	#WRD.74,-(SP)				
23190	102740	012746	010256		MOV	#THR.FMT,-(SP)				
23191	102744	012746	000004		MOV	#4,-(SP)				
23192	102750	010600			MOV	SP,R0	:	SP,*		
23193	102752	104414			TRAP	14				
23194	102754	010316			MOV	R3,(SP)	:	CHANNEL,*		1013
23195	102756	012746	000072		MOV	#72,-(SP)				
23196	102762	012746	007754		MOV	#FMT.21,-(SP)				
23197	102766	012746	000003		MOV	#3,-(SP)				
23198	102772	010600			MOV	SP,R0	:	SP,*		
23199	102774	104414			TRAP	14				
23200	102776	062706	000020		ADD	#20,SP	:			1013
23201	103002	062702	000044		ADD	#44,R2	:	*.BIT.OFFSET		1012
23202	103006	020227	000264		CMP	R2,#264	:	BIT.OFFSET,*		
23203	103012	003722			BLE	17\$				
23204	103014	005001			CLR	R1	:	WRD.CNT		1014
23205	103016	010102			MOV	R1,R2	:	WRD.CNT,*		1014
23206	103020	006302			ASL	R2				
23207	103022	005762	014022		TST	IO.BUF(R2)				
23208	103026	001445			BEQ	21\$				
23209	103030	005267	112342		INC	P.CNT	:			1014
23210	103034	026767	112336	112336	CMP	P.CNT,LIMIT				
23211	103042	003043			BGT	22\$				
23212	103044	104455			TRAP	55	:			1014
23213	103046	000207			.WORD	207				
23214	103050	012750			.WORD	SYNC				
23215	103052	026302			.WORD	DUMPER				
23216	103054	012746	011462		MOV	#WRD.67,-(SP)	:			1014
23217	103060	012746	010650		MOV	#WRD.12,-(SP)				
23218	103064	012746	011552		MOV	#WRD.75,-(SP)				
23219	103070	012746	011540		MOV	#WRD.74,-(SP)				
23220	103074	012746	011416		MOV	#WRD.61,-(SP)				

```

23222      ;ML4AD
23223      ;
23224      ;
23225 103100 012746 010304      MOV    #FIV.FMT,-(SP)
23226 103104 012746 000006      MOV    #6,-(SP)
23227 103110 010600              MOV    SP,R0                ; SP,*
23228 103112 104414              TRAP   14
23229 103114 010316              MOV    R3,(SP)              ; CHANNEL,*
23230 103116 012746 000072      MOV    #7,-(SP)
23231 103122 012746 007754      MOV    #FMT.21,-(SP)
23232 103126 012746 000003      MOV    #3,-(SP)
23233 103132 010600              MOV    SP,R0                ; SP,*
23234 103134 104414              TRAP   14
23235 103136 062706 000024      ADD    #24,SP
23236 103142 005201 21$:      INC    R1                    ; WRD.CNT
23237 103144 020127 000014      CMP    R1,#14                ; WRD.CNT,*
23238 103150 003722              BLE    20$
23239 103152 022626 22$:      CMP    (SP)+,(SP)+
23240 103154 104467              TRAP   67
23241 103156 006000              ROR    R0
23242 103160 103602              BHIS   24$
23243 103162 000167 177352      23$:  JMP    14$
23244 103166 005203 24$:      INC    R3                    ; CHANNEL
23245 103170 020327 000043      CMP    R3,#43                ; CHANNEL,*
23246 103174 003772              BLE    23$
23247 103176 005726              TST    (SP)+
23248 103200 000207              RTS    PC

```

; Routine Size: 309 words
; Maximum stack depth per invocation: 19 words

```

23249
23250
23251
23256
23257
23261
23262      .SBTTL T51 TEST CODE SECTION
23266 103202
23267 103202 004767 176622      T51:: JSR    PC,$T51
23268 103206 104466              TRAP   66
23269 103210 006000              ROR    R0
23270 103212 103773              BLO    1$
23271 103214 000207              RTS    PC

```

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

```

23272
23273
23274

```


23283
23284
23285 :

10155 !<BLF/PAGE>

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (104)

```

23287 :ML4AD
23288 :
23289 :
23290 : 10156 !
23291 : 10157 BGNTST;
23292 : 10158
23293 : 10159 !++
23294 : 10160 TEST NUMBER: TST 52
23295 : 10161
23296 : 10162 TEST NAME: UNCORRECTABLE ERROR SYNDROME DECODE TEST AT CHANNELS > 35
23297 : 10163
23298 : 10164 TEST DESCRIPTION:
23299 : 10165 TEST SYNDROME DECODE TO DETECT BUT
23300 : 10166 NOT CORRECT UNCORRECTABLE CHANNEL
23301 : 10167 ERRORS BY:
23302 : 10168
23303 : 10169 1. VIA ECC DIAG MODE FORCE SYNDROME
23304 : 10170 BITS TO INDICATE CHANNEL ERRORS
23305 : 10171 AT WORDS OF A CRC GROUP BUT AT CHANNELS
23306 : 10172 GREATER THAN 35.
23307 : 10173
23308 : 10174 2. DO A MASS BUS READ TRANSFER
23309 : 10175
23310 : 10176 3. THEN EXAMIN THE IO BUF FOR NO
23311 : 10177 BITS COMPLIMENTED AND ECH AND
23312 : 10178 BITS SET
23313 : 10179
23314 : 10180 IMPLICIT INPUTS:
23315 : 10181 IO_BUF
23316 : 10182 A VECTOR OF 256 WORDS WHERE DATA
23317 : 10183 FOR MBUS READS AND WRITES TRANSFERS
23318 : 10184 IS FOUND.
23319 : 10185
23320 : 10186
23321 : 10187 --
23322 : 10188
23323 : 10189 Local
23324 : 10190 PLOG;
23325 : 10191
23326 : 10192 CLR_THRESHOLD;
23327 : 10193 BAI = ONE;
23328 : 10194 IO_BUF = ZEROES;
23329 : 10195 GD_BLK_XFER ();
23330 : 10196 MLCS1 = write;
23331 : 10197 TIME_OUT_LOOP;
23332 : 10198
23333 : 10199 incr PLOG_SEL from 0 to 6 do
23334 : 10200 begin
23335 : 10201
23336 : 10202 case .PLOG_SEL from 0 to 6 of
23337 : 10203 set
23338 : 10204
23339 : 10205 [0] :
23340 : 10206 PLOG = 0;
23341 : 10207

```

!INDEX INTO REMAINDER TABLE 'REM_TBL'

!CLEAR ERROR PRINT THRESHOLD
!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

29-Mar-1982 16:23:0. TOPS-20 Bliss-16 V2(212)
 29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (104)

```

23343 :ML4AD
23344 :
23345 :
23346 : 10208 [1] : !POINTS TO WORD 1 IN THE CRC GROUP
23347 : 10209 PLOG = 1;
23348 : 10210
23349 : 10211 [2] : !POINTS TO WORD 2 IN THE CRC GROUP
23350 : 10212 PLOG = 2;
23351 : 10213
23352 : 10214 [3] : !POINTS TO WORD 3 IN THE CRC GROUP
23353 : 10215 PLOG = 3;
23354 : 10216
23355 : 10217 [4] : !POINTS TO WORD 4 IN THE CRC GROUP
23356 : 10218 PLOG = 4;
23357 : 10219
23358 : 10220 [5] : !POINTS TO WORD 5 IN THE CRC GROUP
23359 : 10221 PLOG = 5;
23360 : 10222
23361 : 10223 [6] : !POINTS TO ALL 6 WORDS IN THE CRC GROUP
23362 : 10224 PLOG = 58;
23363 : 10225 tes;
23364 : 10226
23365 : 10227 incr CHANNEL from 36 to 62 do
23366 : 10228 begin
23367 : 10229 BGNSUB; !START OF SCOPE LOOP
23368 : 10230 CLR_MBUS; !CLEAR THE DRIVE
23369 : 10231
23370 : 10232 incr COUNT from 0 to 255 do
23371 : 10233 IO_BUF [.COUNT] = ZEROES; !CLEAR THE FIRST CRC GROUP IN THE IO_BUF
23372 : 10234
23373 : 10235 FORCE REM (.PLOG, .CHANNEL); !FORCE UNCORRECTABLE ERROR AT THIS WORD AND CHANNEL
23374 : 10236 GD BLR_XFER (); !SET UP A GOOD BLOCK TRANSFER
23375 : 10237 ECC_DIS = ZERO; !CLEAR ECC DISABLE
23376 : 10238 MLCS1 = read; !DO A READ TRANSFER
23377 : 10239 TIME_OUT_LOOP; !WAIT FOR THE TRANSFER TO COMPLETE
23378 : 10240
23379 : 10241 if not ((.ECH_ERR) and (.UNC_ERR)) !SEE IF THE ERROR WAS DETECTED
23380 : 10242 then
23381 : 10243 begin !ERROR IF NOT DETECTED
23382 : 10244 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23383 : 10245 ERRDF (136, SYNC, DUMPER);
23384 : 10246 PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_76, WRD_9);
23385 : 10247 PRINTB (FMT_21, .PLOG, .CHANNEL);
23386 : 10248 end;
23387 : 10249
23388 : 10250 incr CNT from 0 to 12 do !SEE IF ANY CORRECTION WAS DONE
23389 : 10251
23390 : 10252 if .IO_BUF [.CNT] neq ZEROES !SEARCH THE IO_BUF FOR ALL ZEROES
23391 : 10253 then
23392 : 10254 begin !REPORT AN ERROR IF ANY CORRECTION WAS DONE
23393 : 10255 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23394 : 10256 ERRDF (137, SYNC, DUMPER);
23395 : 10257 PRINTB (FIV_FMT, WRD_61, WRD_74, WRD_75, WRD_12, WRD_67);
23396 : 10258 PRINTB (FMT_21, .PLOG, .CHANNEL);
23397 : 10259 end;
  
```



```

23511      :ML4AD
23512      :
23513      :
23514 103650 003042      BGT      20$
23515 103652 104455      TRAP     55
23516 103654 000211      .WORD   211
23517 103656 012750      .WORD   SYNC
23518 103660 026302      .WORD   DUMPER
23519 103662 012746 011462      MOV     #WRD.67,-(SP)
23520 103666 012746 010650      MOV     #WRD.12,-(SP)
23521 103672 012746 011552      MOV     #WRD.75,-(SP)
23522 103676 012746 011540      MOV     #WRD.74,-(SP)
23523 103702 012746 011416      MOV     #WRD.61,-(SP)
23524 103706 012746 010304      MOV     #FIV.FMT,-(SP)
23525 103712 012746 000006      MOV     #6,-(SP)
23526 103716 010600      MOV     SP,R0
23527 103720 104414      TRAP    14
23528 103722 010416      MOV     R4,(SP)
23529 103724 010346      MOV     R3,-(SP)
23530 103726 012746 007754      MOV     #FMT.21,-(SP)
23531 103732 012746 000003      MOV     #3,-(SP)
23532 103736 010600      MOV     SP,R0
23533 103740 104414      TRAP    14
23534 103742 062706 000024      ADD     #24,SP
23535 103746 005201 19$:      INC     R1
23536 103750 020127 000014      CMP     R1,#14
23537 103754 003723      BLE     18$
23538 103756 022626 20$:      CMP     (SP)+,(SP)+
23539 103760 104467      TRAP    67
23540 103762 006000      ROR     R0
23541 103764 103002      BHIS   22$
23542 103766 000167 177370 21$:      JMP     12$
23543 103772 005204 22$:      INC     R4
23544 103774 020427 000076      CMP     R4,#76
23545 104000 003772      BLE     21$
23546 104002 005205      INC     R5
23547 104004 020527 000006      CMP     R5,#6
23548 104010 003002      BGT     23$
23549 104012 000167 177244      JMP     2$
23550 104016 000207 23$:      RTS     PC
23551
23552
23553
23558
23559
23563
23564
  
```

```

: Routine Size: 193 words
: Maximum stack depth per invocation: 18 words
  
```

.SBTTL T52 TEST CODE SECTION

23572 104020
23573 104020 004767 177172
23574 104024 104466
23575 104026 006000
23576 104030 103773
23577 104032 000207

T52::
T5: JSR PC,\$T52
TRAP 66
ROR R0
BLO T5
RTS PC

1026

23578
23579 ; Routine Size: 6 words
23580 ; Maximum stack depth per invocation: 0 words
23585
23586
23587 ; 10267 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (105)

```

23589 :ML4AD
23590 : TEST CODE SECTION
23591 :
23592 : 10268 !
23593 : 10269 BGNTST;
23594 : 10270
23595 : 10271 !++
23596 : 10272 TEST NUMBER: TST 53
23597 : 10273
23598 : 10274 TEST NAME: MULTIPLE CHANNEL ERROR SYNDROME DECODE TEST
23599 : 10275
23600 : 10276 TEST DESCRIPTION:
23601 : 10277 TEST SYNDROME DECODE TO DETECT BUT NOT
23602 : 10278 CORRECT UNCORRECTABLE MULTIPLE CHANNEL
23603 : 10279 ERRORS BY:
23604 : 10280
23605 : 10281 1. VIA ECC DIAGNOSTIC MODE FORCE SYNDROME
23606 : 10282 BITS TO INDICATE MULTIPLE CHANNEL ERRORS
23607 : 10283 BY ASSIGNING CRC_A EQUAL TO CRC_B
23608 : 10284
23609 : 10285 2. DO A MASS BUS READ TRANSFER
23610 : 10286
23611 : 10287 3. THEN EXAMIN THE IO BUF FOR NO BITS
23612 : 10288 COMPLIMENTED AND ECH AND UNC BITS SET
23613 : 10289
23614 : 10290 IMPLICIT INPUTS:
23615 : 10291 IO_BUF
23616 : 10292 A VECTOR OF 256 WORDS WHERE DATA
23617 : 10293 FOR MBUS READS AND WRITES TRANSFERS
23618 : 10294 IS FOUND.
23619 : 10295
23620 : 10296
23621 : 10297 !--
23622 : 10298
23623 : 10299 Local
23624 : 10300 PLOG; !INDEX INTO REMAINDER TABLE 'REM_TBL'
23625 : 10301
23626 : 10302 CLR_THRESHOLD; !CLEAR ERROR PRINT THRESHOLD
23627 : 10303 BAI = ONE; !FIRST LETS CLEAR THE ML11'S GOOD
23628 : 10304 IO_BUF = ZEROES; !BLOCK BY WRITING ZEROES TO IT
23629 : 10305 GD_BLK_XFER ();
23630 : 10306 MLCS1 = write;
23631 : 10307 TIME_OUT_LOOP;
23632 : 10308
23633 : 10309 incr PLOG_SEL from 0 to 6 do !TEST SYNDROME DECODE AT 7 DIFFERENT PLOG VALUES
23634 : 10310 begin
23635 : 10311
23636 : 10312 case .PLOG_SEL from 0 to 6 of !SELECT A PLOG VALUE
23637 : 10313 set
23638 : 10314
23639 : 10315 [0] : !POINTS TO CRC WORD 0
23640 : 10316 PLOG = 0;
23641 : 10317
23642 : 10318 [1] : !POINTS TO CRC WORD 1
23643 : 10319 PLOG = 1;

```


29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (105)

```

23645 :ML4AD
23646 :
23647 :
23648 : 10320
23649 : 10321 [2] : !POINTS TO CRC WORD 2
23650 : 10322 PLOG = 2;
23651 : 10323
23652 : 10324 [3] : !POINTS TO CRC WORD 3
23653 : 10325 PLOG = 3;
23654 : 10326
23655 : 10327 [4] : !POINTS TO CRC WORD 4
23656 : 10328 PLOG = 4;
23657 : 10329
23658 : 10330 [5] : !POINTS TO CRC WORD 5
23659 : 10331 PLOG = 5;
23660 : 10332
23661 : 10333 [6] : !POINTS TO ALL 6 WORDS IN CRC GROUP
23662 : 10334 PLOG = 58;
23663 : 10335 tes;
23664 : 10336
23665 : 10337 incr CHANNEL from 1 to 35 do !TEST 35 CHANNELS STARTING AT CHANNEL 1
23666 : 10338 begin
23667 : 10339 AGNSUB; !START OF SCOPE LOOP
23668 : 10340 CLR_M:US; !CLEAR THE DRIVE
23669 : 10341
23670 : 10342 incr COUNT from 0 to 255 do !CLEAR FIRST CRC GROUP IN THE IO_BUF
23671 : 10343 IO_BUF [.COUNT] = ZEROES;
23672 : 10344
23673 : 10345 FORCE_REM (.PLOG, .CHANNEL); !FORCE ERR AT THIS WORD AND CHANNEL
23674 : 10346 CRC_A = .CRC_B; !CRC_A EQL TO CRC_B CAUSES THE UNC ERROR
23675 : 10347 GD_BLK_XFER ?); !SET UP A GOOD BLOCK TRANSFER
23676 : 10348 ECC_DIS = ZERO; !CLEAR ECC DISABLE
23677 : 10349 MLC51 = read; !DO A READ TRANSFER
23678 : 10350 TIME_OUT_LOOP; !WAIR UNTIL THE TRANSFER IS COMPLETE
23679 : 10351
23680 : 10352 if not ((.ECH_ERR) and (.UNC_ERR)) !SEE IF THE ERROR WAS DETECTED
23681 : 10353 then
23682 : 10354 begin !REPORT ERROR IF NOT DETECTED
23683 : 10355 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23684 : 10356 ERRDF (138, SYNC, DUMPER);
23685 : 10357 PRINTB (FOR_FMT, WRD_77, WRD_10, WRD_76, WRD_9);
23686 : 10358 PRINTB (FMT_21, .PLOG, .CHANNEL);
23687 : 10359 end;
23688 : 10360
23689 : 10361 incr CNT from 0 to 12 do !SEE IF ANY ERROR CORRECTION WAS DONE
23690 : 10362
23691 : 10363 if .IO_BUF [.CNT] neq ZEROES !TEST THE IO_BUF FOR ALL ZEROES
23692 : 10364 then
23693 : 10365 begin !ERROR IF ANY BITS GOT FLIPPED
23694 : 10366 CMP THRESHOLD; !COMPARE ERROR PRINT THRESHOLD
23695 : 10367 ERRDF (139, SYNC, DUMPER);
23696 : 10368 PRINTB (FIV_FMT, WRD_61, WRD_74, WRD_75, WRD_12, WRD_67);
23697 : 10369 PRINTB (FMT_21, .PLOG, .CHANNEL);
23698 : 10370 end;
23699 : 10371

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (105)

!END OF SCOPE LOOP

```

23701 :ML4AD
23702 :
23703 :
23704 : 10372 ENDSUB;
23705 : 10373 end:
23706 : 10374
23707 : 10375 end:
23708 : 10376
23709 : 10377 ENDTST;
23713 :
23714 :

```

23718	104034	004167	100552		\$T53:	.SBTTL	\$T53 TEST CODE SECTION		
23719	104040	005067	111332			JSR	R1,\$SAVE5	:	1026
23720	104044	152777	000010	111372		CLR	P,CNT	:	1030
23721	104052	005067	107744			BISB	#10,@ML.REG+40	:	1030
23722	104056	004767	114230			CLR	IO.BUF	:	1030
23723	104062	012777	000061	111314		JSR	PC,GD,BLK.XFER	:	1030
23724	104070	105777	111360		1\$:	MOV	#61,@ML.REG	:	1030
23725	104074	100375				TSTB	@ML.REG+50	:	
23726	104076	005005				BPL	1\$:	
23727	104100	010504			2\$:	CLR	R5	: PLOG.SEL	1030
23728	104102	006304				MOV	R5,R4	: PLOG.SEL,*	1031
23729	104104	066407	104110			ASL	R4	:	
23730	104110	000016			3\$:	ADD	3\$(R4),PC	:	
23731	104112	000022				.WORD	4\$-3\$:	
23732	104114	000030				.WORD	5\$-3\$:	
23733	104116	000036				.WORD	6\$-3\$:	
23734	104120	000044				.WORD	7\$-3\$:	
23735	104122	000052				.WORD	8\$-3\$:	
23736	104124	000060				.WORD	9\$-3\$:	
23737	104126	005003			4\$:	CLR	R3	: PLOG	1031
23738	104130	000421				BR	11\$:	1031
23739	104132	012703	000001		5\$:	MOV	#1,R3	:*,PLOG	1031
23740	104136	000416				BR	11\$:	1031
23741	104140	012703	000002		6\$:	MOV	#2,R3	:*,PLOG	1032
23742	104144	000413				BR	11\$:	1031
23743	104146	012703	000003		7\$:	MOV	#3,R3	:*,PLOG	1032
23744	104152	000410				BR	11\$:	1031
23745	104154	012703	000004		8\$:	MOV	#4,R3	:*,PLOG	1032
23746	104160	000405				BR	11\$:	1031
23747	104162	012703	000005		9\$:	MOV	#5,R3	:*,PLOG	1033
23748	104166	000402				BR	11\$:	1031
23749	104170	012703	000072		10\$:	MOV	#72,R3	:*,PLOG	1033
23750	104174	012704	000001		11\$:	MOV	#1,R4	:*,CHANNEL	1033
23751	104200	104402			12\$:	TRAP	2	:	1033
23752	104202	152777	000040	111234		BISB	#40,@ML.REG+40	:	1033
23753	104210	016702	111616			MOV	ML,DUT,R2	:	1033
23754	104214	042702	177770			BIC	#177770,R2	:	1033
23755	104220	142777	000007	111216		BICB	#7,@ML.REG+40	:	1033

23869
23870
23871
23872
23873
23877 104660
23878 104660 004767 177150
23879 104664 104466
23880 104666 006000
23881 104670 103773
23882 104672 000207
23883
23884
23885
23890
23891
23892 ;

:ML4AD
:
TEST CODE SECTION
:SBTTL T53 TEST CODE SECTION
T53::
1\$: JSR PC,ST53
TRAP 66
ROR R0
BLO 1\$
RTS PC

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

1037

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

10378 !<BLF/PAGE>

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (106)

23894 :ML4AD
 23895 :
 23896 :
 23897 :
 23898 :
 23899 :
 23900 :
 23901 :
 23902 :
 23903 :
 23904 :
 23905 :
 23906 :
 23907 :
 23908 :
 23909 :
 23910 :
 23911 :
 23912 :
 23913 :
 23914 :
 23915 :
 23916 :
 23917 :
 23918 :
 23919 :
 23920 :
 23921 :
 23922 :
 23923 :
 23924 :
 23925 :
 23926 :
 23927 :
 23928 :
 23929 :
 23930 :
 23931 :
 23932 :
 23933 :
 23934 :
 23935 :
 23936 :
 23937 :
 23938 :
 23939 :
 23940 :
 23941 :
 23942 :
 23943 :
 23944 :
 23945 :
 23946 :
 23947 :
 23948 :

TEST CODE SECTION

10379 !
 10380 BGNTST;

10381

10382 !++

10383 TEST NUMBER: TST 54

10384

10385 TEST NAME: SINGLE BIT ERROR SYNDROME GENERATION & DECODE TEST

10386

10387 TEST DESCRIPTION:

10388

10389

10390

10391

10392

10393

10394

10395

10396

10397

10398

10399

10400

10401

10402

10403

10404

10405

10406

10407

10408

10409

10410

10411

10412

10413

10414

10415 !--

10416

10417 local

10418

10419

10420

10421

10422

10423 CLR_THRESHOLD;

10424

10425

10426

10427

10428

10429

10430

10380 BGNTST;

!++

TEST NUMBER: TST 54

TEST NAME: SINGLE BIT ERROR SYNDROME GENERATION & DECODE TEST

TEST DESCRIPTION:

TEST SYNDROME GENERATION, SYNDROME DECODE
 AND ERROR CORRECTION FOR SINGLE BIT CHANNEL
 ERRORS BY:

1. CLEAR THE FIRST CRC GROUP IN THE
 IO_BUF AND GOOD BLOCK TO ZEROES
2. VIA DATA DIAGNOSTIC MODE FLIP A BIT
 TO A ONE
3. THEN DO A MASS BUS READ TRANSFER AND
 EXAMIN THE IO BUF FOR ZEROES INDICATING
 THE CORRECTION WAS PERFORMED
4. REPEAT WITH ALL BITS IN THE CRC GROUP

IMPLICIT INPUTS:

IO_BUF
 A VECTOR OF 256 WORDS WHERE DATA
 FOR MBUS READS AND WRITES TRANSFERS
 IS FOUND.

PD_TEMP
 A BIT VECTOR OF 16 BITS WHERE THE READ
 FROM DATA IS STORED AND ACCESSED FROM.

local

DONE,
 NIB_PAT : bitvector [4],
 NIB_SEL,
 GD_WRD_CNT;

CLR_THRESHOLD;

incr WRD_CNT from 0 to 5 do
 begin

incr BIT_CNT from 0 to 35 do
 begin

!DONE FLAG
 !STORAGE FOR SBE DATA GENERATED
 !POINTER WHERE SBE IS TO BE WRITTEN
 !COUNT OF WHERE GOOD 'NIB_SEL' ARE FOUND

!CLEAR ERROR PRINT THRESHOLD

!FORCE SINGLE BIT ERRORS IN ONE CRC GROUP

!FORCE SBE'S AT EACH BIT OF A WORD

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (106)

```

23950 :ML4AD
23951 :
23952 :
23953 :      10431      incr CNT from 0 to 255 do
23954 :      10432      IO_BUF [.CNT] = ZEROES;
23955 :      10433
23956 :      10434      CLR MBUS;
23957 :      10435      GD_BLK_XFER ();
23958 :      10436      MLC51 = write;
23959 :      10437      TIME_OUT_LOOP;
23960 :      10438      CLR MBUS;
23961 :      10439      DONE = ZERO;
23962 :      10440      GD_WRD_CNT = -1;
23963 :      10441      MLD1 = ZEROES;
23964 :      10442      MLD2 = ZEROES;
23965 :      10443      MLE2 = %0'060000';
23966 :      10444      NIB_SEL = .BIT_CNT/4;
23967 :      10445      DAT_DM_XFER ();
23968 :      10446      MLC51 = write;
23969 :      10447      DELAY (ONE_US);
23970 :      10448
23971 :      10449      do
23972 :      10450      begin
23973 :      10451      PD_TEMP = .MLPD;
23974 :      10452
23975 :      10453      if .PD_TEMP [.NIB_SEL] IS_SET
23976 :      10454      then
23977 :      10455      DAT_CLK = ONE
23978 :      10456      else
23979 :      10457      begin
23980 :      10458      GD_WRD_CNT = .GD_WRD_CNT + 1;
23981 :      10459
23982 :      10460
23983 :      10461      if .GD_WRD_CNT eql .WRD_CNT then DONE = ONE else DAT_CLK = ONE;
23984 :      10462
23985 :      10463      end;
23986 :      10464
23987 :      10465      end
23988 :      10466      until .DONE IS_SET;
23989 :      10467
23990 :      10468      NIB_PAT = ZEROES;
23991 :      10469      NIB_PAT [.BIT_CNT mod 4] = ONE;
23992 :      10470      D1_TEMP = ZEROES;
23993 :      10471      D2_TEMP = ZEROES;
23994 :      10472      E2_TEMP = %0'060000';
23995 :      10473      LD_LNG_WRD (.NIB_SEL, .NIB_PAT);
23996 :      10474      WRT_LNG_WRD;
23997 :      10475      DAT_CLK = ONE;
23998 :      10476      DELAY (ONE_US);
23999 :      10477      BGNSUB;
24000 :      10478      CLR MBUS;
24001 :      10479      GD_BLK_XFER ();
24002 :      10480      ECC_DIS = ZERO;
24003 :      10481      MLC51 = read;
24004 :      10482      TIME_OUT_LOOP;

```

```

!CLEAR FIRST CRC GROUP OF IO_BUF
!CLEAR THE DRIVE
!SET UP A GOOF 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

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (106)

```

24006 :ML4AD
24007 :
24008 :
24009 : 10483
24010 : 10484      incr CNT from 0 to 13 do
24011 : 10485
24012 : 10486      if .IO_BUF [.CNT] neq ZEROES
24013 : 10487      then
24014 : 10488      begin
24015 : 10489      CMP THRESHOLD;
24016 : 10490      ERRDF (140, SYNC, DUMPER);
24017 : 10491      PRINTB (THR_FMT, PHR_12, WRD_76, WRD_9);
24018 : 10492      PRINTB (FMT_22, .WRD_CNT, .BIT_CNT);
24019 : 10493      end;
24020 : 10494
24021 : 10495      ENDSUB;
24022 : 10496      end;
24023 : 10497
24024 : 10498      end;
24025 : 10499
24026 : 10500      ENDTST;
24030 :
24031 :

```

```

!SEE IF THE SBE WAS CORRECTED
!TEST THE IO_BUF FOR ALL ZEROES
!ERROR IF THE ERROR WAS NOT CORRECTED
!COMPARE ERROR PRINT THRESHOLD
!END OF SCOPE LOOP

```

24035	104674	004167	077712	ST54:	.SBTTL	ST54 TEST CODE SECTION		
24036	104700	162706	000010		JSR	R1, \$SAVE5	:	1037
24037	104704	005067	110466		SUB	#10, SP	:	
24038	104710	005001			CLR	P, CNT	:	1042
24039	104712	005002		1\$:	CLR	R1	:	1042
24040	104714	005003		2\$:	CLR	R2	:	1042
24041	104716	010304		3\$:	CLR	R3	:	1043
24042	104720	006304			MOV	R3, R4	:	1043
24043	104722	005064	014022		ASL	R4	:	
24044	104726	005203			CLR	IO_BUF (R4)	:	
24045	104730	020327	000377		INC	R3	:	1043
24046	104734	003770			CMP	R3, #377	:	
24047	104736	152777	000040	110500	BLE	3\$:	
24048	104744	016704	111062		BISB	#40, @ML.REG+40	:	1043
24049	104750	042704	177770		MOV	ML_DUT, R4	:	
24050	104754	142777	000007	110462	BIC	#177770, R4	:	
24051	104762	150477	110456		BICB	#7, @ML.REG+40	:	
24052	104766	004767	113320		BISB	R4, @ML.REG+40	:	
24053	104772	012777	000061	110404	JSR	PC, GD.BLK.XFER	:	1043
24054	105000	105777	110450		MOV	#61, @ML.REG	:	1043
24055	105004	100375		4\$:	TSTB	@ML.REG+50	:	
24056	105006	152777	000040	110430	BPL	4\$:	
24057	105014	016704	111012		BISB	#40, @ML.REG+40	:	1043
24058	105020	042704	177770		MOV	ML_DUT, R4	:	
24059	105024	142777	000007	110412	BIC	#177770, R4	:	
24060	105032	150477	110406		BICB	#7, @ML.REG+40	:	
					BISB	R4, @ML.REG+40	:	

						29-Mar-1982 16:23:04	TOPS
						29-Mar-1982 16:21:03	PA:<
24062					:ML4AD		
24063					:		
24064						TEST CODE SECTION	
24065	105036	005066	000002		CLR 2(SP)	: DONE	1043
24066	105042	012716	177777		MOV #-1,(SP)	: *,GD.WRD.CNT	1044
24067	105046	005077	110522		CLR @ML.REG+170	:	1044
24068	105052	005077	110522		CLR @ML.REG+200	:	1044
24069	105056	012777	060000	110500	MOV #60000,@ML.REG+160	:	1044
24070	105064	010246			MOV R2,-(SP)	: BIT.CNT,*	1044
24071	105066	012746	000004		MOV #4,-(SP)	:	
24072	105072	004767	077350		JSR PC,BLS DIV	:	
24073	105076	010005			MOV R0,R5	: *,NIB.SEL	
24074	105100	004767	111306		JSR PC,DAT.DM.XFER	:	1044
24075	105104	012777	000061	110272	MOV #61,@ML.REG	:	1044
24076	105112	012703	000001		MOV #1,R3	: *,SSTMP2	1044
24077	105116	001411			5\$: BEQ 8\$:	
24078	105120	016704	074772		MOV L\$DLY,R4	: *,SSTMP1	
24079	105124	001404			6\$: BEQ 7\$:	
24080	105126	005066	000012		CLR 12(SP)	: SSTMP	
24081	105132	005304			DEC R4	: SSTMP1	
24082	105134	001374			BNE 6\$:	
24083	105136	005303			7\$: DEC R3	: SSTMP2	
24084	105140	000766			BR 5\$:	
24085	105142	010504			8\$: MOV R5,R4	: NIB.SEL,*	1045
24086	105144	006204			ASR R4	:	
24087	105146	006204			ASR R4	:	
24088	105150	006204			ASR R4	:	
24089	105152	062704	015342		ADD #PD.TEMP,R4	:	
24090	105156	017767	110452	110156	9\$: MOV @ML.REG+230,PD.TEMP	:	1045
24091	105164	010446			MOV R4,-(SP)	:	1045
24092	105166	010546			MOV R5,-(SP)	: NIB.SEL,*	
24093	105170	042716	177770		BIC #177770,(SP)	:	
24094	105174	012746	000001		MOV #1,-(SP)	:	
24095	105200	005046			CLR -(SP)	:	
24096	105202	004767	076426		JSR PC,BLSGT2	:	
24097	105206	062706	000010		ADD #10,SP	:	
24098	105212	005300			DEC R0	:	
24099	105214	001411			BEQ 10\$:	
24100	105216	005266	000004		INC 4(SP)	: GD.WRD.CNT	1045
24101	105222	026601	000004		CMP 4(SP),R1	: GD.WRD.CNT,WRD.CNT	1046
24102	105226	001004			BNE 10\$:	
24103	105230	012766	000001	000006	MOV #1,6(SP)	: *,DONE	
24104	105236	000403			BR 11\$:	
24105	105240	152777	000020	110256	10\$: BISB #20,@ML.REG+120	:	
24106	105246	026627	000006	000001	11\$: CMP 6(SP),#1	: DONE,*	1046
24107	105254	001340			BNE 9\$:	
24108	105256	005066	000010		CLR 10(SP)	: NIB.PAT	1046
24109	105262	010246			MOV R2,-(SP)	: BIT.CNT,*	1046
24110	105264	012746	000004		MOV #4,-(SP)	:	
24111	105270	004767	077164		JSR PC,BLSMOD	:	
24112	105274	010004			MOV R0,R4	:	
24113	105276	006200			ASR R0	:	
24114	105300	006200			ASR R0	:	
24115	105302	006200			ASR R0	:	
24116	105304	012703	000014		MOV #14,R3	:	

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Page
24118							
24119							
24120							
24121	105310	060603			ADD SP,R3	: NIB.PAT,*	
24122	105312	060300			ADD R3,R0		
24123	105314	010016			MOV R0,(SP)		
24124	105316	010446			MOV R4, -(SP)		
24125	105320	042716	177770		BIC #177770,(SP)		
24126	105324	012746	000001		MOV #1, -(SP)		
24127	105330	011646			MOV (SP), -(SP)		
24128	105332	004767	076534		JSR PC,BL\$PU2		
24129	105336	005067	106032		CLR D1.TEMP		
24130	105342	005067	106030		CLR D2.TEMP		1047
24131	105346	012767	060000	106024	MOV #60000,E2.TEMP		1047
24132	105354	010516			MOV R5,(SP)	: NIB.SEL,*	1047
24133	105356	016646	000022		MOV 22(SP), -(SP)	: NIB.PAT,*	1047
24134	105362	004767	113766		JSR PC,LD.LNG.WRD		
24135	105366	016777	106002	110200	MOV D1.TEMP,@ML.REG+170		
24136	105374	016777	105776	110202	MOV D2.TEMP,@ML.REG+200		
24137	105402	016777	105772	110154	MOV E2.TEMP,@ML.REG+160		
24138	105410	152777	000020	110106	BISB #20,@ML.REG+120		
24139	105416	012703	000001		MOV #1,R3	: *,\$STMP2	1047
24140	105422	001411			BEO 15\$		1047
24141	105424	016704	074466		MOV LSDLY,R4	: *,\$STMP1	
24142	105430	001404			BEO 14\$		
24143	105432	005066	000026		CLR 26(SP)	: \$STMP	
24144	105436	005304			DEC R4	: \$STMP1	
24145	105440	001374			BNE 13\$		
24146	105442	005303			DEC R3	: \$STMP2	
24147	105444	000766			BR 12\$		
24148	105446	104402			TRAP 2		
24149	105450	152777	000040	107766	BISB #40,@ML.REG+40		1047
24150	105456	016704	110350		MOV ML.DUT,R4		
24151	105462	042704	177770		BIC #177770,R4		
24152	105466	142777	000007	107750	BICB #7,@ML.REG+40		
24153	105474	150477	107744		BISB R4,@ML.REG+40		
24154	105500	004767	112606		JSR PC,GD.BLK.XFER		
24155	105504	142777	000002	110012	BICB #2,@ML.REG+120		1047
24156	105512	012777	000071	107664	MOV #71,@ML.REG		1048
24157	105520	105777	107730		TSTB @ML.REG+50		1048
24158	105524	100375			BPL 16\$		
24159	105526	005003			CLR R3	: CNT	1048
24160	105530	010304			MOV R3,R4	: CNT,*	1048
24161	105532	006304			ASL R4		
24162	105534	005764	014022		TST IO.BUF(R4)		
24163	105540	001440			BEO 18\$		
24164	105542	005267	107630		INC P.CNT		
24165	105546	026767	107624	107624	CMP P.CNT,LIMIT		1048
24166	105554	003036			BGT 19\$		
24167	105556	104455			TRAP 55		1049
24168	105560	000214			.WORD 214		
24169	105562	012750			.WORD SYNC		
24170	105564	026302			.WORD DUMPER		
24171	105566	012746	010616		MOV #WRD.9, -(SP)		
24172	105572	012746	011562		MOV #WRD.76, -(SP)		1049

24174				:ML4AD				
24175				:	TEST CODE SECTION			
24176								
24177	105576	012746	012036		MOV	#PHR.12,-(SP)		
24178	105602	012746	010256		MOV	#THR.FMT,-(SP)		
24179	105606	012746	000004		MOV	#4,-(SP)		
24180	105612	010600			MOV	SP,R0		
24181	105614	104414			TRAP	14	:	SP,*
24182	105616	010216			MOV	R2,(SP)	:	BIT.CNT,*
24183	105620	010146			MOV	R1,-(SP)	:	WRD.CNT,*
24184	105622	012746	010026		MOV	#FMT.22,-(SP)		
24185	105626	012746	000003		MOV	#3,-(SP)		
24186	105632	010600			MOV	SP,R0		
24187	105634	104414			TRAP	14	:	SP,*
24188	105636	062706	000020		ADD	#20,SP		
24189	105642	005203		18\$:	INC	R3	:	CNT
24190	105644	020327	000015		CMP	R3,#15	:	CNT,*
24191	105650	003727			BLE	17\$		
24192	105652	104467		19\$:	TRAP	67	:	
24193	105654	006000			ROR	R0		
24194	105656	103673			BLO	15\$		
24195	105660	062706	000020		ADD	#20,SP		
24196	105664	005202			INC	R2	:	BIT.CNT
24197	105666	020227	000043		CMP	R2,#43	:	BIT.CNT,*
24198	105672	003002			BGT	20\$		
24199	105674	000167	177014		JMP	2\$		
24200	105700	005201		20\$:	INC	R1	:	WRD.CNT
24201	105702	020127	000005		CMP	R1,#5	:	WRD.CNT,*
24202	105706	003002			BGT	21\$		
24203	105710	000167	176776		JMP	1\$		
24204	105714	062706	000010	21\$:	ADD	#10,SP	:	
24205	105720	000207			RTS	PC		
24206								
24207								
24208								
24213								
24214								
24218								
24219								
24223	105722				.SBTTL	T54 TEST CODE SECTION		
24224	105722	004767	176746	T54::	JSR	PC,\$T54	:	
24225	105726	104466		1\$:	TRAP	66		
24226	105730	006000			ROR	R0		
24227	105732	103773			BLO	1\$		
24228	105734	000207			RTS	PC		

: Routine Size: 267 words
 : Maximum stack depth per invocation: 26 words

1049
 1048
 1048
 1049
 1042
 1042
 1042
 1037

24230
24231
24232
24233
24234
24235
24240
24241
24242 ;

:ML4AD

TEST CODE SECTION

29-Mar-1982 16:23:04 TCPS
29-Mar-1982 16:21:03 PA:<

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

10501 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (107)

24244 :ML4AD

TEST CODE SECTION

24245 :
24246 :
24247 : 10502 !
24248 : 10503 BGNTST;
24249 : 10504
24250 : 10505 !++
24251 : 10506 TEST NUMBER: TST 55
24252 : 10507
24253 : 10508 TEST NAME: MULTIPLE BIT ERROR SYNDROME GENERATION AND DECODE TEST
24254 : 10509
24255 : 10510 TEST DESCRIPTION:
24256 : 10511 TEST SYNDROME GENERATION, SYNDROME DECODE
24257 : 10512 AND ERROR CORRECTION FOR MULTIPLE BIT
24258 : 10513 CHANNEL ERRORS BY:
24259 : 10514
24260 : 10515 1. CLEAR THE FIRST CRC GROUP IN THE
24261 : 10516 IO_BUF AND GOOD BLOCK TO ZEROES
24262 : 10517
24263 : 10518 2. VIA DATA DIAG MODE FLIP ALL BITS IN
24264 : 10519 A CHANNEL TO ONES
24265 : 10520
24266 : 10521 3. THEN DO A MASS BUS READ TRANSFER AND
24267 : 10522 EXAMIN THE IO BUF FOR ZEROES INDICATING
24268 : 10523 THE CORRECTION WAS PERFORMED
24269 : 10524
24270 : 10525 4. REPEAT FOR ALL CHANNELS IN THE CRC GROUP
24271 : 10526
24272 : 10527 IMPLICIT INPUTS:
24273 : 10528 PD TEMP
24274 : 10529 A BIT VECTOR OF 16 BITS WHERE THE READ
24275 : 10530 PROM DATA IS STORED OAND ACCESSED FROM.
24276 : 10531
24277 : 10532 IO_BUF
24278 : 10533 A VECTOR OF 256 WORDS WHERE DATA
24279 : 10534 FOR MBUS READS AND WRITES TRANSFERS
24280 : 10535 IS FOUND.
24281 : 10536
24282 : 10537
24283 : 10538 !--
24284 : 10539
24285 : 10540 local
24286 : 10541 NIB_PAT : bitvector [4],
24287 : 10542 NIB_SEL,
24288 : 10543 GD_WRD_CNT;
24289 : 10544
24290 : 10545 CLR_THRESHOLD;
24291 : 10546 E2_TEMP<12, 3> = %b'110';
24292 : 10547
24293 : 10548 incr BIT_CNT from 0 to 35 do
24294 : 10549 begin
24295 : 10550 CLR_MBUS;
24296 : 10551
24297 : 10552 incr CNT from 0 to 255 do
24298 : 10553 IO_BUF [.CNT] = ZEROES;

10502 !
10503 BGNTST;
10504
10505 !++
10506 TEST NUMBER: TST 55
10507
10508 TEST NAME: MULTIPLE BIT ERROR SYNDROME GENERATION AND DECODE TEST
10509

10510 TEST DESCRIPTION:
10511 TEST SYNDROME GENERATION, SYNDROME DECODE
10512 AND ERROR CORRECTION FOR MULTIPLE BIT
10513 CHANNEL ERRORS BY:
10514
10515 1. CLEAR THE FIRST CRC GROUP IN THE
10516 IO_BUF AND GOOD BLOCK TO ZEROES
10517
10518 2. VIA DATA DIAG MODE FLIP ALL BITS IN
10519 A CHANNEL TO ONES
10520
10521 3. THEN DO A MASS BUS READ TRANSFER AND
10522 EXAMIN THE IO BUF FOR ZEROES INDICATING
10523 THE CORRECTION WAS PERFORMED
10524
10525 4. REPEAT FOR ALL CHANNELS IN THE CRC GROUP
10526

10527 IMPLICIT INPUTS:
10528 PD TEMP
10529 A BIT VECTOR OF 16 BITS WHERE THE READ
10530 PROM DATA IS STORED OAND ACCESSED FROM.
10531
10532 IO_BUF
10533 A VECTOR OF 256 WORDS WHERE DATA
10534 FOR MBUS READS AND WRITES TRANSFERS
10535 IS FOUND.
10536

10537
10538 !--
10539
10540 local
10541 NIB_PAT : bitvector [4],
10542 NIB_SEL,
10543 GD_WRD_CNT;
10544
10545 CLR_THRESHOLD;
10546 E2_TEMP<12, 3> = %b'110';
10547
10548 incr BIT_CNT from 0 to 35 do
10549 begin
10550 CLR_MBUS;
10551
10552 incr CNT from 0 to 255 do
10553 IO_BUF [.CNT] = ZEROES;

!STORAGE FOR SBE TO BE GENERATED
!POINTER TO WHERE ERROR IS TO BE WRITTEN
!COUNT OF WHERE GOOD 'NIB_SEL' ARE FOUND

!CLEAR ERROR PRINT THRESHOLD
!LOAD DATA STRUCTURE WITH ZEROES CRC PATTERN

!TEST FOR MULTIPLE ERRORS AT 36 CHANNELS

!CLEAR THE DRIVE

!CLEAR FIRST CRC GROUP IN THE IO_BUF

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 v2(212)
29-Mar-1982 16:21:03 PA:<NE/LE>ML4AD.BLI.4 (107)

```

24300 :ML4AD
24301 :
24302 :
24303 :
24304 :
24305 :
24306 :
24307 :
24308 :
24309 :
24310 :
24311 :
24312 :
24313 :
24314 :
24315 :
24316 :
24317 :
24318 :
24319 :
24320 :
24321 :
24322 :
24323 :
24324 :
24325 :
24326 :
24327 :
24328 :
24329 :
24330 :
24331 :
24332 :
24333 :
24334 :
24335 :
24336 :
24337 :
24338 :
24339 :
24340 :
24341 :
24342 :
24343 :
24344 :
24345 :
24346 :
24347 :
24348 :
24349 :
24350 :
24351 :
24352 :
24353 :
24354 :

```

```

TEST CODE SECTION

10554
10555 GD_BLK_XFER ();
10556 MLC51 = write;
10557 TIME_OUT_LOOP;
10558 GD_WRD_CNT = ZERO;
10559 D1_TEMP = ZEROES;
10560 D2_TEMP = ZEROES;
10561 NIB_PAT = ZEROES;
10562 NIB_SEL = .BIT_CNT/4;
10563 NIB_PAT [.BIT_CNT mod 4] = ONE;
10564 LD [NG_WRD (.NIB_SEL, .NIB_PAT)];
10565 WRT_LNG_WRD;
10566 DAT_DM_XFER ();
10567 MLC51 = write;
10568 DELAY (ONE_US);
10569
10570 do
10571   begin
10572     PD_TEMP = .MLPD;
10573
10574     if .PD_TEMP [.NIB_SEL] IS_SET
10575     then
10576       DAT_CLK = ONE
10577     else
10578       begin
10579         GD_WRD_CNT = .GD_WRD_CNT + 1;
10580         DAT_CLR = ONE;
10581       end;
10582
10583     end
10584 until .GD_WRD_CNT eql 6;
10585
10586 BGNSUB;
10587 CLR_MBUS;
10588 GD_BLK_XFER ();
10589 ECC_DIS = 0;
10590 MLC51 = read;
10591 TIME_OUT_LOOP;
10592
10593 incr WRD_CNT from 0 to 13 do
10594   if .IO_BUF [.WRD_CNT] neq ZEROES
10595   then
10596     begin
10597       CMP_THRESHOLD;
10598       ERRDF (141, SYNC, DUMPER);
10599       PRINTB (THR_FMT, PHR_13, WRD_76, WRD_9);
10600       PRINTB (FMT_22, .WRD_CNT, .BIT_CNT);
10601     end;
10602
10603 ENDSUB;
10604 end;
10605

```

```

!ALSO CLEAR THTE 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 FROM_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
!COMPARE ERROR PRINT THRESHOLD

!END OF SCOPE LOOP

```

24356 :ML4AD
24357 :
24358 :
24359 : 10606
24360 : 10607

TEST CODE SECTION

ENDTST:

Address	Hex	Hex	Hex	Label	Instruction	Comment	Address
24364					.SBTTL	\$T55 TEST CODE SECTION	
24369	105736	004167	076650	\$T55:	JSR	R1,\$SAVE5	1050
24370	105742	024646			CMP	-(SP),-(SP)	
24371	105744	005067	107426		CLR	P,CNT	1054
24372	105750	042767	070000	105422	BIC	#70000,E2.TEMP	1054
24373	105756	052767	060000	105414	BIS	#60000,E2.TEMP	
24374	105764	005003			CLR	R3	: BIT.CNT
24375	105766	152777	000040	107450	1\$: BISB	#40,@ML.REG+40	1054
24376	105774	016702	110032		MOV	ML,DUT,R2	
24377	106000	042702	177770		BIC	#177770,R2	
24378	106004	142777	000007	107432	BICB	#7,@ML.REG+40	
24379	106012	150277	107426		BISB	R2,@ML.REG+40	
24380	106016	005001			CLR	R1	: CNT
24381	106020	010102		2\$:	MOV	R1,R2	: CNT,*
24382	106022	006302			ASL	R2	
24383	106024	005062	014022		CLR	IO.BUF(R2)	
24384	106030	005201			INC	R1	: CNT
24385	106032	020127	000377		CMP	R1,#377	: CNT,*
24386	106036	003770			BLE	2\$	
24387	106040	004767	112246		JSR	PC,GD.BLK.XFER	
24388	106044	012777	000061	107332	MOV	#61,@ML.REG	
24389	106052	105777	107376	3\$:	TSTB	@ML.REG+50	
24390	106056	100375			BPL	3\$	
24391	106060	005005			CLR	R5	: GD.WRD.CNT
24392	106062	005067	105306		CLR	D1.TEMP	
24393	106066	005067	105304		CLR	D2.TEMP	
24394	106072	005016			CLR	(SP)	: NIB.PAT
24395	106074	010346			MOV	R3,-(SP)	: BIT.CNT,*
24396	106076	012746	000004		MOV	#4,-(SP)	
24397	106102	004767	076340		JSR	PC,BLSDIV	
24398	106106	010004			MOV	R0,R4	: *,NIB.SEL
24399	106110	010346			MOV	R3,-(SP)	: BIT.CNT,*
24400	106112	012746	000004		MOV	#4,-(SP)	
24401	106116	004767	076336		JSR	PC,BL\$MOD	
24402	106122	010002			MOV	R0,R2	
24403	106124	006200			ASR	R0	
24404	106126	006200			ASR	R0	
24405	106130	006200			ASR	R0	
24406	106132	012701	000010		MOV	#10,R1	
24407	106136	060601			ADD	SP,R1	: NIB.PAT,*
24408	106140	060100			ADD	R1,R0	
24409	106142	010016			MOV	R0,(SP)	
24410	106144	010246			MOV	R2,-(SP)	

Address	Op Code	Op 2	Op 3	Op 4	Label	Instruction	Comments	Page
24412						BIC	#177770,(SP)	
24413						MOV	#1,-(SP)	
24414						MOV	(SP),-(SP)	
24415	106146	042716	177770			JSR	PC,BLS\$PU2	
24416	106152	012746	000001			MOV	R4,(SP)	
24417	106156	011646				MOV	16(SP),-(SP)	: NIB.SEL,*
24418	106160	004767	075706			JSR	PC,LD.LNG.WRD	: NIB.PAT,*
24419	106164	010416				MOV	D1.TEMP,@ML.REG+170	
24420	106166	016646	000016			MOV	D2.TEMP,@ML.REG+200	
24421	106172	004767	113156			MOV	E2.TEMP,@ML.REG+160	
24422	106176	016777	105172	107370		JSR	PC,DAT.DM.XFER	
24423	106204	016777	105166	107372		MOV	#61,@ML.REG	
24424	106212	016777	105162	107344		MOV	#1,R1	
24425	106220	004767	110166			MOV	#1,R1	: *,\$STMP2
24426	106224	012777	000061	107152		BEQ	7\$	
24427	106232	012701	000001			MOV	LSDLY,R2	: *,\$STMP1
24428	106236	001411				BEQ	6\$	
24429	106240	016702	073652			CLR	22(SP)	: \$STMP
24430	106244	001404				DEC	R2	: \$STMP1
24431	106246	005066	000022			BNE	5\$	
24432	106252	005302				DEC	R1	: \$STMP2
24433	106254	001374				BR	4\$	
24434	106256	005301				MOV	R4,R2	: NIB.SEL,*
24435	106260	000766				ASR	R2	
24436	106262	010402				ASR	R2	
24437	106264	006202				ASR	R2	
24438	106266	006202				ADD	#PD.TEMP,R2	
24439	106270	006202				MOV	@ML.REG+230,PD.TEMP	
24440	106272	062702	015342			MOV	R2,-(SP)	
24441	106276	017767	107332	107036	8\$:	MOV	R4,-(SP)	: NIB.SEL,*
24442	106304	010246				BIC	#177770,(SP)	
24443	106306	010446				MOV	#1,-(SP)	
24444	106310	042716	177770			CLR	-(SP)	
24445	106314	012746	000001			JSR	PC,BLS\$GT2	
24446	106320	005046				ADD	#10,SP	
24447	106322	004767	075306			DEC	R0	
24448	106326	062706	000010			BEQ	9\$	
24449	106332	005300				INC	R5	: GD.WRD.CNT
24450	106334	001401				BISB	#20,@ML.REG+120	: GD.WRD.CNT,*
24451	106336	005205				CMP	R5,#6	
24452	106340	152777	000020	107156	9\$:	BNE	8\$	
24453	106346	020527	000006			TRAP	2	
24454	106352	001351				BISB	#40,@ML.REG+40	
24455	106354	104402			10\$:	MOV	ML,DUT,R2	
24456	106356	152777	000040	107060		BIC	#177770,R2	
24457	106364	016702	107442			BICB	#7,@ML.REG+40	
24458	106370	042702	177770			BISB	R2,@ML.REG+40	
24459	106374	142777	000007	107042		JSR	PC,GD.BLK.XFER	
24460	106402	150277	107036			BICB	#2,@ML.REG+120	
24461	106406	004767	111700			MOV	#71,@ML.REG	
24462	106412	142777	000002	107104		TSTB	@ML.REG+50	
24463	106420	012777	000071	106756		BPL	11\$	
24464	106426	105777	107022		11\$:	CLR	R1	: WRD.CNT
24465	106432	100375						
24466	106434	005001						

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

Address	OpCode	Operand 1	Operand 2	Label	Instruction	Comments	Address
24468							
24469							
24470							
24471	106436	010102		12\$:	MOV R1,R2	: WRD.CNT,*	1059
24472	106440	006302			ASL R2		
24473	106442	005762	014022		TST ID.BUF(R2)		
24474	106446	00144C			BEQ 13\$		
24475	106450	005267	106722		INC P.CNT	:	1059
24476	106454	026767	106716 106716		CMP P.CNT,LIMIT		
24477	106462	003036			BGT 14\$		
24478	106464	104455			TRAP 55	:	1059
24479	106466	000215			.WORD 215		
24480	106470	012750			.WORD SYNC		
24481	106472	026302			.WORD DUMPER		
24482	106474	012746	010616		MOV #WRD.9,-(SP)	:	1060
24483	106500	012746	011562		MOV #WRD.76,-(SP)		
24484	106504	012746	012060		MOV #PHR.13,-(SP)		
24485	106510	012746	010256		MOV #THR.FMT,-(SP)		
24486	106514	012746	000004		MOV #4,-(SP)		
24487	106520	010600			MOV SP,R0	: SP,*	
24488	106522	104414			TRAP 14		
24489	106524	010316			MOV R3,(SP)	: BIT.CNT,*	1060
24490	106526	010146			MOV R1,-(SP)	: WRD.CNT,*	
24491	106530	012746	010026		MOV #FMT.22,-(SP)		
24492	106534	012746	000003		MOV #3,-(SP)		
24493	106540	010600			MOV SP,R0	: SP,*	
24494	106542	104414			TRAP 14		
24495	106544	062706	000020		ADD #20,SP		1059
24496	106550	005201		13\$:	INC R1	: WRD.CNT	1059
24497	106552	020127	000015		CMP R1,#15	: WRD.CNT,*	
24498	106556	003727			BLE 12\$		
24499	106560	104467		14\$:	TRAP 67	:	1060
24500	106562	006000			ROR R0		
24501	106564	103673			BLO 10\$		
24502	106566	062706	000020		ADD #20,SP		1054
24503	106572	005203			INC R3	: BIT.CNT	1054
24504	106574	020327	000043		CMP R3,#43	: BIT.CNT,*	
24505	106600	003002			BGT 15\$		
24506	106602	000167	177160		JMP 1\$		
24507	106606	022626		15\$:	CMP (SP)+,(SP)+	:	1050
24508	106610	000207			RTS PC		
24509							
24510							
24511							
24516							
24517							
24521							
24522							

: Routine Size: 214 words
 : Maximum stack depth per invocation: 24 words

.SBTTL T55 TEST CODE SECTION

24524
24525
24526
24530 106612
24531 106612 004767 177120
24532 106616 104466
24533 106620 006000
24534 106622 103773
24535 106624 000207
24536
24537
24538
24543
24544
24545 ;

:ML4AD
:
TEST CODE SECTION

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

T55::
1\$: JSR PC,\$T55 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

1060

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

10608 !<BLF/PAGE>

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (108)

24547 :ML4AD
24548 :
24549 :
24550 :
24551 :
24552 :
24553 :
24554 :
24555 :
24556 :
24557 :
24558 :
24559 :
24560 :
24561 :
24562 :
24563 :
24564 :
24565 :
24566 :
24567 :
24568 :
24569 :
24570 :
24571 :
24572 :
24573 :
24574 :
24575 :
24576 :
24577 :
24578 :
24579 :
24580 :
24581 :
24582 :
24583 :
24584 :
24585 :
24586 :
24587 :
24588 :
24589 :
24590 :
24591 :
24592 :
24593 :
24594 :
24595 :
24596 :
24597 :
24598 :
24599 :
24600 :
24601 :

TEST CODE SECTION

10609 !
10610 BGNST;

10611
10612 !++

10613 ! TEST NUMBER: TST 56

10614
10615 ! TEST NAME: ECC ERROR REGISTER TEST

10616
10617 ! TEST DESCRIPTION:

10618 ! TEST THE ECC ERROR REGISTER FOR
10619 ! CLEARING AND LATCHING OF ECC ERROR
10620 ! INFORMATION ON DETECTION
10621 ! OF ECC ERRORS BY:

- 10622 1. THIS TEST IS TABLE DRIVEN. DATA TABLE (DT_1)
10623 ! CONTAINS ERROR FORCING DATA AND EXPECTED
10624 ! MLEE DATA AS A RESULT OF THE FORCED ERROR
- 10625 2. THE TEST THEREFOR INDEXES INTO DT_1 TO FORCE
10626 ! AN ECC ERROR
- 10627 3. A MASS BUS READ TRANSFER IS DONE
- 10628 4. DT_1 IS AGAIN INDEXED AND ITS CONTENTS
10629 ! IS COMPARED AGAINST THE RESULTING
10630 ! MLEE REGISTER CONTENTS
- 10631 5. A MASS BUS CLEAR IS DONE AN MLEE IS CHECKED FOR
10632 ! ZEROES
- 10633 6. THE ECH BIT IS ALSO EXAMINED FOR CORRECT
10634 ! LATCHING INFORMATION

10635
10636
10637
10638
10639
10640
10641
10642
10643
10644 !--

10645
10646 local
10647 ! TMP E1,
10648 ! FINISH,
10649 ! ERR_FLG;

10650
10651 CLR MBUS;
10652 FINISH = 4;

10653
10654 !incr CLR_LOOP from 0 to 1 do
10655 !THE SECOND PASS TESTS MLEE FOR CLEARING
10656 !begin

10657
10658 !incr LATCH_LOOP from 0 to .FINISH do
10659 !begin
10660 !BGNSUB;

!TEMP STORAGE FOR E1 DATA WORD
!VARIABLE ENDING CONDITION FOR LATCH_LOOP
!ERROR FLAG

!CLEAR THE DRIVE
!MAKE LATCH_LOOP DO 5 LOOPS AT FIRST

!THE FIRST PASS TESTS MLEE FOR LATCHING

!TEST MLEE FOR LATCHING BY USING DT_1 DATA

!START OF SCOPE LOOP

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (108)

24603 :ML4AD
24604 :
24605 :
24606 :
24607 :
24608 :
24609 :
24610 :
24611 :
24612 :
24613 :
24614 :
24615 :
24616 :
24617 :
24618 :
24619 :
24620 :
24621 :
24622 :
24623 :
24624 :
24625 :
24626 :
24627 :
24628 :
24629 :
24630 :
24631 :
24632 :
24633 :
24634 :
24635 :
24636 :
24637 :
24638 :
24639 :
24640 :
24641 :
24642 :
24643 :
24644 :
24645 :
24646 :
24647 :
24648 :
24649 :
24650 :
24651 :
24652 :
24653 :
24654 :
24655 :
24656 :
24657 :

TEST CODE SECTION

```

ERR_FLG = ZERO;           !CLEAR THE ERROR FLAG
MLER = ZEROES;           !CLEAR THE ERROR REGISTER
ECC_DM = ONE;            !ENABLE ECC DIAG MODE
+
THE FOLLOWING ASSIGNMENT:
    MLE1 = .TMP_E1
IS EQUIVALENT TO THE FOLLOWING
TWO ASSINGMENTS:
    PAR_CRC_WRD = .DT_1 [.LATCH_LOOP, PO_5];           !LOAD PAR_CRC WRD WITH REM_TBL
    CRC_A = .DT_1 [.LATCH_LOOP, AO_5];                 !LOAD CRC_A WITH REM_TBL
THIS IS NECESSARY DUE TO THE FACT THAT
THE E1 REGISTER IS WORD ORIENTATED AND
THE BLISS COMPILER GENERATES BYTE INST
TO ACCESS THE REGISTER.
-
TMP_E1 = ZEROES;           !CLEAR OUT THE TEMP WORD
TMP_E1<0, 6> = .DT_1 [.LATCH_LOOP, AO_5];           !LOAD CRC_A WITH REM_TBL
TMP_E1<8, 6> = .DT_1 [.LATCH_LOOP, PO_5];           !LOAD PAR_CRC WRD WITH REM_TBL
MLET = .TMP_E1;           !LOAD THE E1 REGISTER WITH DATA
CRC_B = .DT_1 [.LATCH_LOOP, BO_5];           !GET CRC B FROM DT 1
GD_BLK_XFER();           !SET UP A GOOD BLOCK TRANSFER
ECC_DIS = ZERO;           !DISABLE ECC DIABLE
MLCS1 = read;           !READ THE ERROR AND LACTH MLEE
TIME_OUT_LOOP;
if .CRC_ERR neq .DT_1 [.LATCH_LOOP, CRC_DATA] then ERR_FLG = ONE;           !TEST CRC_ERR
if .SGL_ERR neq .DT_1 [.LATCH_LOOP, SGL_DATA] then ERR_FLG = ONE;           !TEST SGL_ERR
if .UNC_ERR neq .DT_1 [.LATCH_LOOP, UNC_DATA] then ERR_FLG = ONE;           !TEST UNC
if .BIT_IN_ERR neq .DT_1 [.LATCH_LOOP, EO_5] then ERR_FLG = ONE;           !TEST BIT IN ERROR
if .CHAN_IN_ERR neq .DT_1 [.LATCH_LOOP, CO_5] then ERR_FLG = ONE;           !TEST CHANNEL IN ERROR
if .ECH_ERR neq .DT_1 [.LATCH_LOOP, ECH_DATA]           !TEST IF ECH GOT SET
then
begin                               !REPORT ERROR IF NOT SET
ERRDF (123, ASYNC, DUMPER);
PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_69, WRD_14);
end;
if .ERR_FLG IS_SET           !TEST IF THE ERROR FLAG GOT SET
then
begin                               !REPORT ERROR IF SET
ERRDF (124, SYNC, DUMPER);

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2:212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (108)

```

24659 :ML4AD
24660 :
24661 :
24662 : 10713 PRINTB (THR_FMT, REG_14, WRD_52, PHR_11);
24663 : 10714 PRINTB (FMT_1, .DT_1 [.LATCH_LOOP, EE_DATA], .MLEE);
24664 : 10715 PRINTB (FMT_19, .DT_1 [.LATCH_LOOP, BO_5], .DT_1 [.LATCH_LOOP, AO_5], .DT_1 [.LATCH_LOOP, PO_5]);
24665 : 10716 end;
24666 : 10717
24667 : 10718 ENDSUB; !END OF SCOPE LOOP
24668 : 10719 end;
24669 : 10720
24670 : 10721 CLR_MBUS; !CLEAR THE REGISTER
24671 : 10722
24672 : 10723 if .MLEE neq ZERO !SEE IF THE REGISTER GOT CLEARED
24673 : 10724 then
24674 : 10725 begin !REPORT ERROR IF NOT CLEARED
24675 : 10726 ERRDF (125, SYNC, DUMPER);
24676 : 10727 PRINTB (FOR_FMT, REG_14, WRD_52, FNC_7, WRD_14);
24677 : 10728 end;
24678 : 10729
24679 : 10730 ECC_DM = ONE; !ENABLE ECC DISABLE
24680 : 10731 FINISH = ZERO; !MAKE LATCH_LOOP LOOP ONCE TO CLEAR MLEE AGAIN
24681 : 10732 end;
24682 : 10733
24683 : 10734 ENDTST;
24687 :
24688 :

```

```

24692 106626 004167 075760 $T56: .SBTTL $T56 TEST CODE SECTION
24693 106632 162706 000006 JSR R1,$$SAVE5 ; 1060
24694 106636 152777 000040 106600 SUB #6,SP ;
24695 106644 016705 107162 BISB #40,@ML.REG+40 ; 1064
24696 106650 042705 177770 MOV ML,DUT,R5
24697 106654 142777 000007 106562 BIC #177770,R5
24698 106662 150577 106556 BISB R5,@ML.REG+40
24699 106666 011766 000004 MOV (PC),4(SP) ; *,FINISH 1065
24700 106672 005066 000002 CLR 2(SP) ; CLR.LOOP 1065
24701 106676 005004 1$: CLR R4 ; LATCH.LOOP 1065
24702 106700 000167 000656 JMP 168
24703 106704 010446 2$: MOV R4,-(SP) ; LATCH.LOOP,* 1068
24704 106706 012746 000006 MOV #6,-(SP)
24705 106712 004767 075304 JSR PC,BLSMUL
24706 106716 010005 MOV R0,R5
24707 106720 012701 015764 MOV #DT_1,R1 ; 1069
24708 106724 060501 ADD R5,R1 ;
24709 106726 012702 015766 MOV #DT_1+2,R2 ; 1068
24710 106732 060502 ADD R5,R2 ;
24711 106734 104402 3$: TRAP 2 ; 1065
24712 106736 005003 CLR R3 ; ERR.FLG 1066
24713 106740 005077 106520 CLR @ML.REG+60 ; 1066

```

Address	OpCode	Op2	Op3	Op4	Label	Instruction	Comments	Page
24715					:ML4AD			
24716					:			
24717						TEST CODE SECTION		
24718	106744	152777	000001	106552		BISB #1,AML.REG+120		
24719	106752	005066	000004			CLR 4(SP)		1066
24720	106756	111200				MOVB (R2),RO	TMP.E1	1068
24721	106760	042700	177700			BIC #177700,RO		1068
24722	106764	142766	000077	000004		BICB #77,4(SP)		
24723	106772	050066	000004			BIS RO,4(SP)	*.TMP.E1	
24724	106776	011200				MOV (R2),RO	*.TMP.E1	
24725	107000	006300				ASL RO		1068
24726	107002	006300				ASL RO		
24727	107004	042700	140377			BIC #140377,RO		
24728	107010	042766	037400	000004		BIC #37400,4(SP)	*.TMP.E1	
24729	107016	050066	000004			BIS RO,4(SP)	*.TMP.E1	
24730	107022	016677	000004	106524		MOV 4(SP),AML.REG+150	TMP.E1,*	1068
24731	107030	116500	015770			MOVB DT,1+4(R5),RO		1068
24732	107034	042700	177700			BIC #177700,RO		
24733	107040	142777	000077	106516		BICB #77,AML.REG+160		
24734	107046	150077	106512			BISB RO,AML.REG+160		
24735	107052	004767	111234			JSR PC,GD.BLK.XFER		1068
24736	107056	142777	000002	106440		BICB #2,AML.REG+120		1068
24737	107064	012777	000071	106312		MOV #71,AML.REG		1068
24738	107072	105777	106356		4\$:	TSTB AML.REG+50		
24739	107076	100375				BPL 4\$		
24740	107100	011146				MOV (R1),-(SP)		1069
24741	107102	042716	157777			BIC #157777,(SP)		
24742	107106	017700	106502			MOV AML.REG+210,RO		
24743	107112	042700	157777			BIC #157777,RO		
24744	107116	020026				CMP RO,(SP)+		
24745	107120	001402				BEQ 5\$		
24746	107122	012703	000001			MOV #1,R3	*.ERR.FLG	
24747	107126	011146			5\$:	MOV (R1),-(SP)		1069
24748	107130	042716	137777			BIC #137777,(SP)		
24749	107134	017700	106454			MOV AML.REG+210,RO		
24750	107140	042700	137777			BIC #137777,RO		
24751	107144	020026				CMP RO,(SP)+		
24752	107146	001402				BEQ 6\$		
24753	107150	012703	000001			MOV #1,R3	*.ERR.FLG	
24754	107154	011146			6\$:	MOV (R1),-(SP)		1069
24755	107156	042716	077777			BIC #77777,(SP)		
24756	107162	017700	106426			MOV AML.REG+210,RO		
24757	107166	042700	077777			BIC #77777,RO		
24758	107172	020026				CMP RO,(SP)+		
24759	107174	001402				BEQ 7\$		
24760	107176	012703	000001			MOV #1,R3	*.ERR.FLG	
24761	107202	111146			7\$:	MOVB (R1),-(SP)		1069
24762	107204	042716	177700			BIC #177700,(SP)		
24763	107210	117700	106400			MOVB AML.REG+210,RO		
24764	107214	042700	177700			BIC #177700,RO		
24765	107220	020026				CMP RO,(SP)+		
24766	107222	001402				BEQ 8\$		
24767	107224	012703	000001			MOV #1,R3	*.ERR.FLG	
24768	107230	011146			8\$:	MOV (R1),-(SP)		1070
24769	107232	042716	170077			BIC #170077,(SP)		

Address	Label	Instruction	Comments	Count
24771				
24772				
24773				
24774	107236	017700 106352	MOV 3ML,REG+210,RO	
24775	107242	042700 170077	BIC #170077,RO	
24776	107246	020026	CMP RO,(SP)+	
24777	107250	001402	BEQ 9S	
24778	107252	012703 000001	MOV #1,R3	
24779	107256	011246	MOV (R2),-(SP)	: *.ERR.FLG
24780	107260	005046	CLR -(SP)	: 1070
24781	107262	032766 010000 000002	BIT #10000,2(SP)	
24782	107270	001401	BEQ 10S	
24783	107272	005216	INC (SP)	
24784	107274	005000	CLR RO	
24785	107276	132777 000100 106160	BITB #100,3ML,REG+60	
24786	107304	001401	BEQ 11S	
24787	107306	005200	INC RO	
24788	107310	020026	CMP RO,(SP)+	
24789	107312	001002	BNE 12S	
24790	107314	005726	TST (SP)+	
24791	107316	000425	BR 13S	
24792	107320	005726	TST (SP)+	
24793	107322	104455	TRAP 55	: 1070
24794	107324	000173	.WORD 173	:
24795	107326	012706	.WORD ASYNC	:
24796	107330	026302	.WORD DUMPER	:
24797	107332	012746 010664	MOV #WORD.14,-(SP)	: 1070
24798	107336	012746 011474	MOV #WORD.69,-(SP)	:
24799	107342	012746 010630	MOV #WORD.10,-(SP)	:
24800	107346	012746 011462	MOV #WORD.67,-(SP)	:
24801	107352	012746 010270	MOV #FOR.FMT,-(SP)	:
24802	107356	012746 000005	MOV #5,-(SP)	:
24803	107362	010600	MOV SP,RO	: SP,*
24804	107364	104414	TRAP 14	:
24805	107366	062706 000014	ADD #14,SP	: 1070
24806	107372	020327 000001	CMP R3,#1	: ERR.FLG,*
24807	107376	001062	BNE 14S	: 1070
24808	107400	104455	TRAP 55	:
24809	107402	000174	.WORD 174	:
24810	107404	012750	.WORD SYNC	:
24811	107406	026302	.WORD DUMPER	:
24812	107410	012746 012020	MOV #PHR.11,-(SP)	: 1071
24813	107414	012746 011310	MOV #WORD.52,-(SP)	:
24814	107420	012746 012620	MOV #REG.14,-(SP)	:
24815	107424	012746 010256	MOV #THR.FMT,-(SP)	:
24816	107430	012746 000004	MOV #4,-(SP)	:
24817	107434	010600	MOV SP,RO	: SP,*
24818	107436	104414	TRAP 14	:
24819	107440	017716 106150	MOV 3ML,REG+210,(SP)	: 1071
24820	107444	011146	MOV (R1),-(SP)	:
24821	107446	012746 006442	MOV #FMT.1,-(SP)	:
24822	107452	012746 000003	MOV #3,-(SP)	:
24823	107456	010600	MOV SP,RO	: SP,*
24824	107460	104414	TRAP 14	:
24825	107462	011200	MOV (R2),RO	: 1071

Address	Hex	Op	Val1	Val2	Op	Val1	Val2	Label	Address
24827									
24828									
24829									
24830	107464	006200			ASR	R0			
24831	107466	006200			ASR	R0			
24832	107470	006200			ASR	R0			
24833	107472	006200			ASR	R0			
24834	107474	006200			ASR	R0			
24835	107476	006200			ASR	R0			
24836	107500	042700	177700		BIC	#177700,R0			
24837	107504	010016			MOV	R0,(SP)			
24838	107506	111246			MOVB	(R2),-(SP)			
24839	107510	042716	177700		BIC	#177700,(SP)			
24840	107514	116546	015770		MOVB	DT.1+4(R5),-(SP)			
24841	107520	042716	177700		BIC	#177700,(SP)			
24842	107524	012746	007632		MOV	#FMT.19,-(SP)			
24843	107530	012746	000004		MOV	#4,-(SP)			
24844	107534	010600			MOV	SP,R0			
24845	107536	104414			TRAP	14		: SP,*	
24846	107540	062706	000030		ADD	#30,SP		:	
24847	107544	104467			TRAP	67		:	1071
24848	107546	006000			ROR	R0		:	1071
24849	107550	103002			BHIS	15\$:	
24850	107552	000167	177156		JMP	3\$:	
24851	107556	022626			CMP	(SP)+,(SP)+		:	1065
24852	107560	005204			INC	R4		:	1065
24853	107562	020466	000004		CMP	R4,4(SP)		:	LATCH.LOOP
24854	107566	003002			BGT	17\$:	LATCH.LOOP,FINISH
24855	107570	000167	177110		JMP	2\$:	
24856	107574	152777	000040	105642	BISB	#40,@ML.REG+40		:	1071
24857	107602	016705	106224		MOV	ML.DUT,R5		:	
24858	107606	042705	177770		BIC	#177770,R5		:	
24859	107612	142777	000007	105624		#7,@ML.REG+40		:	
24860	107620	150577	105620			R5,@ML.REG+40		:	
24861	107624	005777	105764			@ML.REG+210		:	1072
24862	107630	001424			BEG	18\$:	
24863	107632	104455			TRAP	55		:	1072
24864	107634	000175			.WORD	175		:	
24865	107636	012750			.WORD	SYNC		:	
24866	107640	026302			.WORD	DUMPER		:	
24867	107642	012746	010664		MOV	#WRD.14,-(SP)		:	1072
24868	107646	012746	012234		MOV	#FNC.7,-(SP)		:	
24869	107652	012746	011310		MOV	#WRD.52,-(SP)		:	
24870	107656	012746	012620		MOV	#REG.14,-(SP)		:	
24871	107662	012746	010270		MOV	#FOR.FMT,-(SP)		:	
24872	107666	012746	000005		MOV	#5,-(SP)		:	
24873	107672	010600			MOV	SP,R0		:	
24874	107674	104414			TRAP	14		:	SP,*
24875	107676	062706	000014		ADD	#14,SP		:	
24876	107702	152777	000001	105614	BISB	#1,@ML.REG+120		:	1072
24877	107710	005046	000004		CLR	4(SP)		:	1073
24878	107714	005266	000002		INC	2(SP)		:	FINISH
24879	107720	026627	000002	000001	CMP	2(SP),#1		:	CLR.LOOP
24880	107726	003002			BGT	19\$:	CLR.LOOP,*
24881	107730	000167	176742		JMP	1\$:	

24883
 24884
 24885
 24886 107734 062706 000006
 24887 107740 000207
 24888
 24889
 24890
 24895
 24896
 24900
 24901
 24905 107742
 24906 107742 004767 176660
 24907 107746 104466
 24908 107750 006000
 24909 107752 103773
 24910 107754 000207
 24911
 24912
 24913
 24918
 24919
 24920 : 10735 !<BLF/PAGE>

```

;ML4AD
;
TEST CODE SECTION
19$:  ADD  #6,SP
      RTS  PC
;
; Routine Size: 294 words
; Maximum stack depth per invocation: 23 words

.SBTTL T56 TEST CODE SECTION
T56::
1$:  JSR  PC,$T56
      TRAP 66
      ROR  R0
      BLO  1$
      RTS  PC
;
; Routine Size: 6 words
; Maximum stack depth per invocation: 0 words
  
```

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<
 1060

1073

24922 :ML4AD
24923 :
24924 :
24925 :
24926 :
24927 :
24928 :
24929 :
24930 :
24931 :
24932 :
24933 :
24934 :
24935 :
24936 :
24937 :
24938 :
24939 :
24940 :
24941 :
24942 :
24943 :
24944 :
24945 :
24946 :
24947 :
24948 :
24949 :
24950 :
24951 :
24952 :
24953 :
24954 :
24955 :
24956 :
24957 :
24958 :
24959 :
24960 :
24961 :
24962 :
24963 :
24964 :
24965 :
24966 :
24967 :
24968 :
24969 :
24970 :
24971 :
24972 :
24973 :
24974 :
24975 :
24976 :

TEST CODE SECTION

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (109)

10736 !
10737 BGNTST;

10738 !++
10739
10740 TEST NUMBER: TST 57

10741
10742 TEST NAME: ECC ERROR LOCATION REGISTER TEST

10743
10744 TEST DESCRIPTION:
10745 TEST THE ECC ERROR LOCATION REGISTER FOR CLOCKING, BIT UNIQUENESS
10746 CLEARING AND LATCHING OF DSA ADDRESSES ON DETECTION OF ECC ERROR BY:
10747

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 AND 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

10777 IMPLICIT INPUTS:
10778 IO_BUF
10779 A VECTOR OF 256 WORDS WHERE DATA
10780 FOR MBUS READS AND WRITES TRANSFERS
10781 IS FOUND.

10782
10783
10784 --
10785
10786 local
10787 SGL,

!SINGLE BIT ERROR PARRAMETER

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (109)

```

24978 :ML4AD
24979 :      TEST CODE SECTION
24980 :
24981 :      10788      UNC,
24982 :      10789      SAVE,
24983 :      10790      SHF_DATA,
24984 :      10791      DSA_PAT;
24985 :      10792
24986 :      10793 CLR_THRESHOLD;
24987 :      10794 BGNSUB;
24988 :      10795 CLR_MBUS;
24989 :      10796 FOR_ECC_ERR (SGL = 1, UNC = 0);
24990 :      10797 MLDA = .LST_BLK;
24991 :      10798 MLWC = not 255;
24992 :      10799 MLBA = IO_BUF;
24993 :      10800 MLCS1 = read;
24994 :      10801 TIME_OUT_LOOP;
24995 :      10802
24996 :      10803 if .MLEL eql ZEROES
24997 :      10804 then
24998 :      10805     begin
24999 :      10806     ERRDF (126, SYNC, DUMPER);
25000 :      10807     PRINTB (THR_FMT, REG_15, WRD_52, PHR_11);
25001 :      10808     EXIT_TST;
25002 :      10809     end;
25003 :      10810
25004 :      10811 ENDSUB;
25005 :      10812 SHF_DATA = .LST_ARR;
25006 :      10813
25007 :      10814 incr LOOP from 0 to 16 do
25008 :      10815     begin
25009 :      10816     BGNSUB;
25010 :      10817     CLR_MBUS;
25011 :      10818     FOR_ECC_ERR (SGL = 1, UNC = 0);
25012 :      10819
25013 :      10820     selectone .LOOP of
25014 :      10821     set
25015 :      10822
25016 :      10823     [0] :
25017 :      10824     begin
25018 :      10825     DSA_PAT = ZEROES;
25019 :      10826     MLDA = .DSA_PAT;
25020 :      10827     end;
25021 :      10828
25022 :      10829     [1] :
25023 :      10830     begin
25024 :      10831     DSA_PAT = .LST_BLK;
25025 :      10832     MLDA = .DSA_PAT;
25026 :      10833     end;
25027 :      10834
25028 :      10835     [otherwise] :
25029 :      10836     begin
25030 :      10837     DSA_PAT = .SHF_DATA;
25031 :      10838     MLDA = .SHF_DATA;
25032 :      10839     SHF_DATA = .SHF_DATA^-ONE;

```

!UNCORRECTABLE ERR PARRAMETER
!TEMPORY SAVE LOCATION
!SHIFTING DATA PATTERN SAVE LOCATION
!EXPECTED DSA PATTERN LATCHED IN MLEL

!CLEAR ERROR PRINT THRESHOLD
!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

29-Mar-1982 16:23:04 TOPS-20 bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (109)

```

25034 :ML4AD
25035 :
25036 :
25037 : 10840         end;
25038 : 10841         tes;
25039 : 10842
25040 : 10843         MLWC = not 255;
25041 : 10844         MLBA = IO_BUF;
25042 : 10845         MLCS1 = read;
25043 : 10846         TIME_OUT_LOOP;
25044 : 10847         SAVE = .MLEL;
25045 : 10848
25046 : 10849         if .SAVE neq .DSA_PAT
25047 : 10850         then
25048 : 10851         begin
25049 : 10852             CMP THRESHOLD;
25050 : 10853             ERRDF (127, ASYNC, DUMPER);
25051 : 10854             PRINTB (TWO_FMT, REG 15, PHR 4);
25052 : 10855             PRINTB (FMT_2, .DSA_PAT, .SAVE, (.DSA_PAT xor .SAVE));
25053 : 10856         end;
25054 : 10857
25055 : 10858         ENDSUB;
25056 : 10859         end;
25057 : 10860
25058 : 10861         CLR_MBUS;
25059 : 10862         FOR_ECC_ERR (SGL = 1, UNC = 0);
25060 : 10863         MLWC = not 255;
25061 : 10864         MLBA = IO_BUF;
25062 : 10865         MLDA = .LST_BLK;
25063 : 10866         MLCS1 = read;
25064 : 10867         TIME_OUT_LOOP;
25065 : 10868
25066 : 10869         incr LOOP from 0 to 4 do
25067 : 10870         begin
25068 : 10871             BGNSUB;
25069 : 10872             MLER = ZEROES;
25070 : 10873
25071 : 10874             case .LOOP from 0 to 4 of
25072 : 10875             set
25073 : 10876             [0] :
25074 : 10877                 begin
25075 : 10878                     SGL = 1;
25076 : 10879                     UNC = 0;
25077 : 10880                     DSA_PAT = .LST_BLK;
25078 : 10881                     MLDA = ZEROES;
25079 : 10882                 end;
25080 : 10883
25081 : 10884             [1] :
25082 : 10885                 begin
25083 : 10886                     SGL = 0;
25084 : 10887                     UNC = 1;
25085 : 10888                     DSA_PAT = ZEROES;
25086 : 10889                     MLDA = ZEROES;
25087 : 10890                 end;
25088 : 10891

```

```

!SET UP THE TRANSFER
!DO THE TRANSFER AND LATCH MLEL
!READ AND SAVE THE MLEL REGISTER
!SEE IF SAVE IS WHAT WE EXPECTED
!REPORT AN ERROR IF NOT THE SAME
!COMPARE ERROR PRINT THRESHOLD
!END OF SCOPE LOOP
!CLEAR THE DRIVE
!FORCE A SBE
!SET UP A LAST BLOCK TRANSFER
!TO LATCH THE MLEL WITH T
!NOW TEST FOR CORRECT LATCHING
!START OF SCOPE LOOP
!CLEAR THE ERROR REGISTER
!SELECT TEST DATA
!FORCE A SBE AND CHECK FOR THE LST BLK ADRS
!FORCE A UNC ERROR AND CHECK FOR ADRS ZERO

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (109)

```

25090 :ML4AD
25091 :
25092 :
25093 : 10892
25094 : 10893
25095 : 10894
25096 : 10895
25097 : 10896
25098 : 10897
25099 : 10898
25100 : 10899
25101 : 10900
25102 : 10901
25103 : 10902
25104 : 10903
25105 : 10904
25106 : 10905
25107 : 10906
25108 : 10907
25109 : 10908
25110 : 10909
25111 : 10910
25112 : 10911
25113 : 10912
25114 : 10913
25115 : 10914
25116 : 10915
25117 : 10916
25118 : 10917
25119 : 10918
25120 : 10919
25121 : 10920
25122 : 10921
25123 : 10922
25124 : 10923
25125 : 10924
25126 : 10925
25127 : 10926
25128 : 10927
25129 : 10928
25130 : 10929
25131 : 10930
25132 : 10931
25133 : 10932
25134 : 10933
25135 : 10934
25136 : 10935
25137 : 10936
25138 : 10937
25139 : 10938
25140 : 10939
25141 : 10940
25142 : 10941
25143 : 10942
25144 : 10943

```

```

TEST CODE SECTION

[2] :
    begin
    SGL = 1;
    UNC = 0;
    DSA PAT = ZEROES;
    MLDA = .LST_BLK;
    end;

[3] :
    begin
    SGL = 0;
    UNC = 1;
    DSA PAT = ZEROES;
    MLDA = .LST_BLK;
    end;

[4] :
    begin
    CLR_MBUS;
    SGL = 1;
    UNC = 0;
    DSA PAT = ZEROES;
    MLDA = .LST_BLK;
    end;
tes;
FOR ECC_ERR (.SGL, .UNC);
MLWC = not 255;
MLBA = IO_BUF;
MLCS1 = read;
TIME_OUT_LOOP;

if .LOOP neq 4
then
    SAVE = .MLEL
else
    begin
    CLR_MBUS;
    SAVE = .MLEL;
    end;

if .SAVE neq .DSA_PAT
then
    begin
    CMP THRESHOLD;
    ERRDF (128, SYNC, DUMPER);
    PRINTB (THR_FMT, REG 15, WRD 71, WRD_10);
    PRINTB (FMT_1, .DSA_PAT, .SAVE)
    end;

ENDSUB;

```

```

!FORCE SBE AND CHECK FOR ADRS ZERO

!FORCE UNC ERROR AND CHECK FOR ADRS ZERO

!FORCE SBE AND CHECK FOR ADRS ZERO

!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
!COMPARE ERROR PRINT THRESHOLD

!END OF SCOPE LOOP

```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 BLISS-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (109)

25146 ;ML4AD
25147 :
25148 :
25149 :
25150 :
25151 :
25155 :
25156 :

TEST CODE SECTION

10944 end;
10945
10946 ENDTST;

Address	Hex	Hex	Hex	Label	Comment	Address
25160	107756	004167	074630	ST57:	.SBTTL ST57 TEST CODE SECTION	
25161	107762	005746			JSR R1,\$SAVES	1073
25162	107764	005067	105406		TST -(SP)	
25163	107770	104402		1\$:	CLR P.CNT	1079
25164	107772	152777	000040		TRAP 2	1079
25165	110000	016705	106026	105444	BISB #40,@ML.REG+40	1079
25166	110004	042705	177770		MOV ML.DUT,R5	
25167	110010	142777	000007	105426	BIC #177770,R5	
25168	110016	150577	105422		BICB #7,@ML.REG+40	
25169	110022	012703	000001		BISB R5,@ML.REG+40	
25170	110026	010346			MOV #1,R3	1079
25171	110030	005001			MOV R3,-(SP)	: *,SGL
25172	110032	005046			CLR R1	: SGL,*
25173	110034	004767	110150		CLR -(SP)	: UNC
25174	110040	016777	103750	105366	JSR PC,FOR.ECC.ERR	
25175	110046	012777	177400	105340	MOV LST.BLK,@ML.REG+30	1079
25176	110054	012777	014022	105342	MOV #-400,@ML.REG+10	1079
25177	110062	012777	000071	105314	MOV #10.BUF,@ML.REG+20	1079
25178	110070	105777	105360	2\$:	MOV #71,@ML.REG	1080
25179	110074	100375			TSTB @ML.REG+50	
25180	110076	005777	105522		BPL 2\$	
25181	110102	001025			TST @ML.REG+220	1080
25182	110104	104455			BNE 3\$	
25183	110106	000176			TRAP 5\$	1080
25184	110110	012750			.WORD 176	
25185	110112	026302			.WORD SYNC	
25186	110114	012746	012020		.WORD DUMPER	
25187	110120	012746	011310		MOV #PHR.11,-(SP)	1080
25188	110124	012746	012626		MOV #WRD.52,-(SP)	
25189	110130	012746	010256		MOV #REG.15,-(SP)	
25190	110134	012746	000004		MOV #THR.FMT,-(SP)	
25191	110140	010600			MOV #4,-(SP)	
25192	110142	104414			MOV SP,R0	: SP,*
25193	110144	104463			TRAP 14	
25194	110146	062706	000016		TRAP 63	
25195	110152	000167	001052		ADD #16,SP	1080
25196	110156	022626		3\$:	JMP 31\$	1080
25197	110160	104467			CMP (SP)+,(SP)+	1079
25198	110162	006000			TRAP 67	1080
25199	110164	103701			ROR R0	
25200	110166	016705	103626		BLO 1\$	
					MOV LST.ARR,R5	: *.SHF.DATA

25202				:ML4AD							29-Mar-1982 16:23:04	TOPS
25203				:		TEST CODE SECTION					29-Mar-1982 16:21:03	PA:<
25204												
25205	110172	005002				CLR R2		:	LOOP			1081
25206	110174	104402			4\$:	TRAP 2		:				1081
25207	110176	152777	000040	105240		BISB #40,@ML.REG+40		:				1081
25208	110204	016700	105622			MOV ML.DUT,R0		:				
25209	110210	042700	177770			BIC #177770,R0		:				
25210	110214	142777	000007	105222		BICB #7,@ML.REG+40		:				
25211	110222	150077	105216			BISB R0,@ML.REG+40		:				
25212	110226	012703	000001			MOV #1,R3		:	*SGL			1081
25213	110232	010346				MOV R3,-(SP)		:	SGL,*			
25214	110234	005001				CLR R1		:	UNC			
25215	110236	005046				CLR -(SP)		:				
25216	110240	004767	107744			JSR PC,FOR.ECC.ERR		:				
25217	110244	005702				TST R2		:	LOOP			1082
25218	110246	001002				BNE 5\$:				
25219	110250	005004				CLR R4		:	DSA.PAT			1082
25220	110252	000405				BR 6\$:				1082
25221	110254	020227	000001		5\$:	CMP R2,#1		:	LOOP,*			1082
25222	110260	001005				BNE 7\$:				
25223	110262	016704	103526			MOV LST.BLK,R4		:	*DSA.PAT			1083
25224	110266	010477	105142		6\$:	MOV R4,@ML.REG+30		:	DSA.PAT,*			1083
25225	110272	000404				BR 8\$:				1082
25226	110274	010504			7\$:	MOV R5,R4		:	SHF.DATA,DSA.PAT			1083
25227	110276	010577	105132			MOV R5,@ML.REG+30		:	SHF.DATA,*			1083
25228	110302	006205				ASR R5		:	SHF.DATA			1083
25229	110304	012777	177400	105102	8\$:	MOV #-400,@ML.REG+10		:				1084
25230	110312	012777	014022	105104		MOV #10,BUF,@ML.REG+20		:				1084
25231	110320	012777	000071	105056		MOV #71,@ML.REG		:				1084
25232	110326	105777	105122		9\$:	TSTB @ML.REG+50		:				
25233	110332	100375				BPL 9\$:				
25234	110334	017766	105264	000004		MOV @ML.REG+220,4(SP)		:	*SAVE			1084
25235	110342	026604	000004			CMP 4(SP),R4		:	SAVE,DSA.PAT			1084
25236	110346	001451				BEQ 11\$:				
25237	110350	005267	105022			INC P.CNT		:				
25238	110354	026767	105016	105016		CMP P.CNT,LIMIT		:				1085
25239	110362	003402				BLE 10\$:				
25240	110364	022626				CMP (SP)+,(SP)+		:				
25241	110366	000445				BR 12\$:				
25242	110370	104455			10\$:	TRAP 55		:				1085
25243	110372	000177				.WORD 177		:				
25244	110374	012706				.WORD ASYNC		:				
25245	110376	026302				.WORD DUMPER		:				
25246	110400	012746	011676			MOV #PHR.4,-(SP)		:				1085
25247	110404	012746	012626			MOV #REG.15,-(SP)		:				
25248	110410	012746	010246			MOV #TWO.FMT,-(SP)		:				
25249	110414	012746	000003			MOV #3,-(SP)		:				
25250	110420	010600				MOV SP,R0		:	SP,*			
25251	110422	104414				TRAP 14		:				
25252	110424	016616	000014			MOV 14(SP),(SP)		:	SAVE,*			1085
25253	110430	010446				MOV R4,-(SP)		:	DSA.PAT,*			
25254	110432	046616	000002			BIC 2(SP),(SP)		:				
25255	110436	040466	000002			BIC R4,2(SP)		:	DSA.PAT,*			
25256	110442	052616				BIS (SP)+,(SP)		:				

Address	Hex	Hex	Hex	Label	Instruction	Comments	Page
25258				:ML4AD			
25259				:	TEST CODE SECTION		
25260							
25261	110444	016646	000014		MOV 14(SP),-(SP)	: SAVE,*	
25262	110450	010446			MOV R4,-(SP)	: DSA.PAT,*	
25263	110452	012746	006506		MOV #FMT.2,-(SP)		
25264	110456	012746	000004		MOV #4,-(SP)		
25265	110462	010600			MOV SP,R0	: SP,*	
25266	110464	104414			TRAP 14		
25267	110466	062706	000020		ADD #20,SP		
25268	110472	022626		11\$:	CMP (SP)+,(SP)+		1085
25269	110474	104467			TRAP 67		1081
25270	110476	006000			ROR R0		1085
25271	110500	103635			BLO 4\$		
25272	110502	005202		12\$:	INC R2	: LOOP	
25273	110504	020227	000020		CMP R2,#20	: LOOP,*	1081
25274	110510	003631			BLE 4\$		
25275	110512	152777	000040	104724	BISB #40,@ML.REG+40		1085
25276	110520	016705	105306		MOV ML.DUT,R5		
25277	110524	042705	177770		BIC #177770,R5		
25278	110530	142777	000007	104706	BICB #7,@ML.REG+40		
25279	110536	150577	104702		BISB R5,@ML.REG+40		
25280	110542	012701	000001		MOV #1,R3	: *,SGL	
25281	110546	010346			MOV R3,-(SP)	: SGL,*	1086
25282	110550	005001			CLR R1	: UNC	
25283	110552	005046			CLR -(SP)		
25284	110554	004767	107430		JSR PC,FOR.ECC.ERR		
25285	110560	012777	177400	104626	MOV #-400,@ML.REG+10		1086
25286	110566	012777	014022	104630	MOV #10.BUF,@ML.REG+20		1086
25287	110574	016777	103214	104632	MOV LST.BLK,@ML.REG+30		1086
25288	110602	012777	000071	104574	MOV #71,@ML.REG		1086
25289	110610	105777	104640	13\$:	TSTB @ML.REG+50		
25290	110614	100375			BPL 13\$		
25291	110616	005002			CLR R2	: LOOP	1086
25292	110620	104402		14\$:	TRAP 2		1087
25293	110622	005077	104636		CLR @ML.REG+60		1087
25294	110626	010205			MOV R2,R5	: LOOP,*	1087
25295	110630	006305			ASL R5		
25296	110632	066507	110636		ADD 15\$(R5),PC		
25297	110636	000012		15\$:	.WORD 16\$-15\$		
25298	110640	000026			.WORD 17\$-15\$		
25299	110642	000044			.WORD 19\$-15\$		
25300	110644	000046			.WORD 20\$-15\$		
25301	110646	000056			.WORD 21\$-15\$		
25302	110650	012703	000001	16\$:	MOV #1,R3	: *,SGL	1087
25303	110654	005001			CLR R1	: UNC	1088
25304	110656	016704	103132		MOV LST.BLK,R4	: *,DSA.PAT	1088
25305	110662	000404			BR 18\$		1088
25306	110664	005003		17\$:	CLR R3	: SGL	1088
25307	110666	012701	000001		MOV #1,R1	: *,UNC	1088
25308	110672	005004			CLR R4	: DSA.PAT	1088
25309	110674	005077	104534	18\$:	CLR @ML.REG+30		1089
25310	110700	000430			BR 24\$		1087
25311	110702	000420		19\$:	BR 22\$		1089
25312	110704	005003		20\$:	CLR R3	: SGL	1090


```
25370      ;ML4AD
25371      ;
25372      ;
25373 111202 022626      28$:  CMP      (SP)+,(SP)+      ;
25374 111204 104467      ; TRAP      67      ;
25375 111206 006000      ; ROR      R0      ;
25376 111210 103603      ; BLO      14$      ;
25377 111212 005202      29$:  INC      R2      ; LOOP
25378 111214 020227 000004 ; CMP      R2,#4      ; LOOP,*
25379 111220 003002      ; BGT      30$      ;
25380 111222 000167 177372 ; JMP      14$      ;
25381 111226 022626      30$:  CMP      (SP)+,(SP)+      ;
25382 111230 005726      31$:  TST      (SP)+      ;
25383 111232 000207      ; RTS      PC      ;
25384
25385      ; Routine Size: 343 words
25386      ; Maximum stack depth per invocation: 19 words
25391
25392
25396
25397      .SBTTL T57 TEST CODE SECTION
25401 111234      T57::
25402 111234 004767 176516 1$:  JSR      PC,$T57      ;
25403 111240 104466      ; TRAP      66      ;
25404 111242 006000      ; ROR      R0      ;
25405 111244 103773      ; BLO      1$      ;
25406 111246 000207      ; RTS      PC      ;
25407
25408      ; Routine Size: 6 words
25409      ; Maximum stack depth per invocation: 0 words
25414
25415
25416 ;      10947 !<BLF/PAGE>
```

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (110)

```

25418 :ML4AD
25419 :
25420 : TEST CODE SECTION
25421 :
25422 : 10948 !
25423 : 10949 BGNTST:
25424 : 10950
25425 : 10951 !++
25426 : 10952 TEST NUMBER: TST 58
25427 : 10953
25428 : 10954 TEST NAME: ECC HARD ERROR BIT TEST
25429 : 10955
25430 : 10956 TEST DESCRIPTION:
25431 : 10957 VIA ECC_DM AND ECC_DIS TEST THE
25432 : 10958 ECH BIT FOR SETTING AND NOT SETTING BY:
25433 : 10959
25434 : 10960 1. VIA ECC_DM AND A WRITE TRANSFER FORCE
25435 : 10961 ECC ERRORS INTO THE ML11'S GOOD BLOCK
25436 : 10962
25437 : 10963 2. WITH ECC_DM = 0 AND ECC_DIS = 0
25438 : 10964 DO A READ TRANSFER AND CHECK ECH SET
25439 : 10965
25440 : 10966 3. WITH ECC_DM = 0 AND ECC_DIS = 1
25441 : 10967 DO A READ TRANSFER AND CHECK ECH NOT SET
25442 : 10968
25443 : 10969 IMPLICIT INPUTS:
25444 : 10970 IO_BUF
25445 : 10971 A VECTOR OF 256 WORDS WHERE DATA
25446 : 10972 FOR MBUS READS AND WRITES TRANSFERS
25447 : 10973 IS FOUND.
25448 : 10974
25449 : 10975
25450 : 10976 --
25451 : 10977
25452 : 10978 Local
25453 : 10979 EXPECTED; !EXPECTED ECH_ERR VALUE
25454 : 10980
25455 : 10981 CLR_MBUS; !CLEAR THE DRIVE
25456 : 10982 ECC_DM = ONE; !SET ECC DIAG MODE
25457 : 10983 PAR_CRC_WRD = ZEROES; !FORCE UNCORRECTABLE ERRORS INTO THE GOOD BLOCK
25458 : 10984 CRC_A = ZEROES;
25459 : 10985 CRC_B = ZEROES;
25460 : 10986 BAI = ONE;
25461 : 10987 IO_BUF = ONES;
25462 : 10988 GD_BLK_XFER ();
25463 : 10989 MLCS1 = write;
25464 : 10990 TIME_OUT_LOOP;
25465 : 10991
25466 : 10992 incr LOOP from 0 to 1 do !TEST ECH BIT FOR SETTING / NOT SETTING
25467 : 10993 begin
25468 : 10994 BGNSUB; !START OF SCOPE LOOP
25469 : 10995 CLR_MBUS; !CLEAR THE DRIVE
25470 : 10996 GD_BLK_XFER (); !SET UP A GOOD BLOCK TRANSFER
25471 : 10997
25472 : 10998 if .LOOP eql ZERO !SELECT THE TEST DATA
25473 : 10999 then

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI 4 (110)

```

25474 :ML4AD
25475 :
25476 :
25477 :      11000      begin
25478 :      11001      ECC_DIS = 0;
25479 :      11002      EXPECTED = 1;
25480 :      11003      end
25481 :      11004      else
25482 :      11005      begin
25483 :      11006      ECC_DIS = 1;
25484 :      11007      EXPECTED = 0;
25485 :      11008      end;
25486 :      11009
25487 :      11010      MLC51 = read;
25488 :      11011      TIME_OUT_LOOP;
25489 :      11012
25490 :      11013      if .ECH_ERR neg .EXPECTED
25491 :      11014      then
25492 :      11015      begin
25493 :      11016      ERRDF (129, SYNC, DUMPER);
25494 :      11017      PRINTB (FOR_FMT, WRD_67, WRD_10, WRD_69, WRD_14);
25495 :      11018      PRINTB (FMT_1, .EXPECTED, .ECH_ERR);
25496 :      11019      end;
25497 :      11020
25498 :      11021      ENDSUB;
25499 :      11022      end;
25500 :      11023
25501 :      11024      ENDTST;

```

```

!EXPECT ECH BIT TO SET
!EXPECT ECH BIT NOT TO SET
!READ THE ERROR OUT
!IS ECH BIT WHAT WE EXPECTED
!ERROR IF NOT
!END OF SCOPE LOOP

```

25506					.SBTTL	\$T58 TEST CODE SECTION		
25510	111250	004167	073300		\$T58:	JSR	R1,\$SAVE3	
25511	111254	152777	000040	104162		BISB	#40,@ML.REG+40	1094
25512	111262	016702	104544			MOV	ML,DUT,R2	1097
25513	111266	042702	177770			BIC	#177770,R2	
25514	111272	142777	000007	104144		BICB	#7,@ML.REG+40	
25515	111300	150277	104140			BISB	R2,@ML.REG+40	
25516	111304	152777	000001	104212		BISB	#1,@ML.REG+120	1098
25517	111312	042777	037400	104234		BIC	#37400,@ML.REG+150	1098
25518	111320	142777	000077	104226		BICB	#77,@ML.REG+150	1098
25519	111326	142777	000077	104230		BICB	#77,@ML.REG+160	1098
25520	111334	152777	000010	104102		BISB	#10,@ML.REG+40	1098
25521	111342	012767	177777	102452		MOV	#-1,IO.BUF	1098
25522	111350	004767	106736			JSR	PC,GD,BLK.XFER	1098
25523	111354	012777	000061	104022		MOV	#61,@ML.REG	1098
25524	111362	105777	104066		1\$:	TSTB	@ML.REG+50	1098
25525	111366	100375				BPL	1\$	
25526	111370	005001				CLR	R1	1099
25527	111372	104402			2\$:	TRAP	2	1099
25528	111374	152777	000040	104042		BISB	#40,@ML.REG+40	1099

Address	Hex	Hex	Hex	Label	Instruction	Comment	Line
25530				:ML4AD			
25531				:			
25532					TEST CODE SECTION		
25533	111402	016700	104424		MOV ML,DUT,RO		
25534	111406	042700	177770		BIC #177770,RO		
25535	111412	142777	000007	104024	BICB #7,@ML,REG+40		
25536	111420	150077	104020		BISB RO,@ML,REG+40		
25537	111424	004767	106662		JSR PC,GD,BLK,XFER		
25538	111430	005701			TST R1	: LOOP	1099
25539	111432	001006			BNE 3\$		1099
25540	111434	142777	000002	104062	BICB #2,@ML,REG+120		
25541	111442	012702	000001		MOV #1,R2	: *.EXPECTED	1100
25542	111446	000404			BR 4\$		1100
25543	111450	152777	000002	104046	3\$: BISB #2,@ML,REG+120		1099
25544	111456	005002			CLR R2	: EXPECTED	1100
25545	111460	012777	000071	103716	4\$: MOV #71,@ML,REG		1100
25546	111466	105777	103762		5\$: TSTB @ML,REG+50		1101
25547	111472	100375			BPL 5\$		
25548	111474	010203			MOV R2,R3	: EXPECTED,*	1101
25549	111476	005000			CLR R0		
25550	111500	132777	000100	103756	BITB #100,@ML,REG+60		
25551	111506	001401			BEQ 6\$		
25552	111510	005200			INC R0		
25553	111512	020003		6\$:	CMP R0,R3		
25554	111514	001441			BEQ 8\$		
25555	111516	104455			TRAP 5\$		1101
25556	111520	000201			.WORD 201		
25557	111522	012750			.WORD SYNC		
25558	111524	026302			.WORD DUMPER		
25559	111526	012746	010664		MOV #WRD.14,-(SP)		1101
25560	111532	012746	011474		MOV #WRD.69,-(SP)		
25561	111536	012746	010630		MOV #WRD.10,-(SP)		
25562	111542	012746	011462		MOV #WRD.67,-(SP)		
25563	111546	012746	010270		MOV #FOR.FMT,-(SP)		
25564	111552	012746	000005		MOV #5,-(SP)		
25565	111556	010600			MOV SP,R0	: SP,*	
25566	111560	104414			TRAP 14		
25567	111562	005016			CLR (SP)		
25568	111564	132777	000100	103672	BITB #100,@ML,REG+60		1101
25569	111572	001401			BEQ 7\$		
25570	111574	005216			INC (SP)		
25571	111576	010246		7\$:	MOV R2,-(SP)	: EXPECTED,*	
25572	111600	012746	006442		MOV #FMT.1,-(SP)		
25573	111604	012746	000003		MOV #3,-(SP)		
25574	111610	010600			MOV SP,R0	: SP,*	
25575	111612	104414			TRAP 14		
25576	111614	062706	000022		ADD #22,SP		1101
25577	111620	104467		8\$:	TRAP 67		1101
25578	111622	006000			ROR R0		
25579	111624	103662			BLO 2\$		
25580	111626	005201			INC R1	: LOOP	1099
25581	111630	020127	000001		CMP R1,#1	: LOOP,*	
25582	111634	003656			BLE 2\$		
25583	111636	000207			RTS PC		1094
25584							

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

25586
25587
25588
25589
25590
25595
25596
25600
25601
25605 111640
25606 111640 004767 177404
25607 111644 104466
25608 111646 006000
25609 111650 103773
25610 111652 000207
25611
25612
25613
25618
25619
25620 :

:ML4AD
:
: TEST CODE SECTION
: Routine Size: 124 words
: Maximum stack depth per invocation: 13 words

.SBTTL T58 TEST CODE SECTION
T58::
1\$: JSR PC,\$T58 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

11025 !<BLF/PAGE>

1102

29-Mar-1982 16:23:04
 29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
 PA:<NEALE>ML4AD.BLI.4 (111)

```

25622 :ML4AD
25623 :      TEST CODE SECTION
25624 :
25625 :      11026 !
25626 :      11027 BGNTST;
25627 :      11028
25628 :      11029 !++
25629 :      11030 TEST NUMBER: TST 59
25630 :      11031
25631 :      11032 TEST NAME:          DATA CHECK ERROR BIT TEST
25632 :      11033
25633 :      11034 TEST DESCRIPTION:
25634 :      11035 VIA ECC_EN, ECC_DM AND ECC_DM TEST THE
25635 :      11036 DCK BIT FOR SETTING AND NOT SETTING BY:
25636 :      11037
25637 :      11038 1. VIA ECC_DM AND A WRITE TRANSFER FORCE
25638 :      11039 ECC ERRORS INTO THE ML11'S GOOD BLOCK
25639 :      11040
25640 :      11041 2. WITH DCK_EN = 1, ECC_DM = 0, ECC_DIS = 0
25641 :      11042 DO A READ TRANSFER AND CHECK THE DCK BIT SET
25642 :      11043
25643 :      11044 3. WITH DCK_EN = 0, ECC_DM = 0, ECC_DIS = 0
25644 :      11045 DO A READ TRANSFER AND CHECK THE DCK BIT NOT SET
25645 :      11046
25646 :      11047 4. WITH ECC_EN = 1, ECC_DM = 0, ECC_DIS = 1
25647 :      11048 DO A READ TRANSFER AND CHECK THE DCK BIT SET
25648 :      11049
25649 :      11050
25650 :      11051 IMPLICIT INPUTS:
25651 :      11052 IO_BUF
25652 :      11053 A VECTOR OF 256 WORDS WHERE DATA
25653 :      11054 FOR MBUS READS AND WRITES TRANSFERS
25654 :      11055 IS FOUND.
25655 :      11056
25656 :      11057
25657 :      11058 !--
25658 :      11059
25659 :      11060 local
25660 :      11061 SGL,
25661 :      11062 UNC,
25662 :      11063 EXPECTED;
25663 :      11064
25664 :      11065 CLR_MBUS;
25665 :      11066 FOR_ECC_ERR (SGL = 1, UNC = 0);
25666 :      11067 BAI = ONE;
25667 :      11068 IO_BUF = ZEROES;
25668 :      11069 GD_BLK_XFER ();
25669 :      11070 MLTST1 = write;
25670 :      11071 TIME_OUT_LOOP;
25671 :      11072
25672 :      11073 incr LOOP from 0 to 2 do
25673 :      11074 begin
25674 :      11075 BGNSUB;
25675 :      11076 CLR_MBUS;
25676 :      11077 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
  
```

25678 :ML4AD
25679 :
25680 :
25681 :
25682 :
25683 :
25684 :
25685 :
25686 :
25687 :
25688 :
25689 :
25690 :
25691 :
25692 :
25693 :
25694 :
25695 :
25696 :
25697 :
25698 :
25699 :
25700 :
25701 :
25702 :
25703 :
25704 :
25705 :
25706 :
25707 :
25708 :
25709 :
25710 :
25711 :
25712 :
25713 :
25714 :
25715 :
25716 :
25717 :
25718 :
25719 :
25720 :
25721 :
25722 :
25723 :
25724 :
25728 :
25729 :

TEST CODE SECTION

case .LOOP from 0 to 2 of
set

[0] :
begin
DCK_EN = 1;
ECC_DIS = 0;
EXPECTED = 1;
end;

[1] :
begin
DCK_EN = 0;
ECC_DIS = 0;
EXPECTED = 0;
end;

[2] :
begin
DCK_EN = 1;
ECC_DIS = 1;
EXPECTED = 1;
end;

tes;

MLCS1 = read;
TIME_OUT_LOOP;

if .DCK neq .EXPECTED
then

begin

if .LOOP neq ONE then ERRDF (130, SYNC, DUMPER) else ERRDF (130, ASYNC, DUMPER);

PRINTB (THR_FMT, WRD_72, PHR_5, WRD_10);
PRINTB (FMT_1, .EXPECTED, .DCK);
end;

ENDSUB;
end;

11121 ENDTST;

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (111)

!SELECT THE TEST DATA

!EXPECT THE DCK BIT TO SET

!EXPECT THE DCK BIT NOT TO SET

!EXPECT THE DCK BIT TO SET

!READ THE ERROR AND CLOCK THE DCK BIT

!IS THE DCK BIT WHAT WE EXPECTED

!REPORT THE ERRRO IF NOT THE SAME

!CHOOSE WHICH THE ERROR IS ON

!END OF THE SCOPE LOOP

.SBTTL \$T59 TEST CODE SECTION

Address	Hex	Hex	Hex	Label	Instruction	Comments	Hex
25790				:ML4AD			
25791				:	TEST CODE SECTION		
25792							
25793	112154	001405			BEG 11\$		
25794	112156	104455			TRAP 55		
25795	112160	000202			.WORD 202		
25796	112162	012750			.WORD SYNC		
25797	112164	026302			.WORD DUMPER		
25798	112166	000404			BR 12\$		
25799	112170	104455		11\$:	TRAP 55		
25800	112172	000202			.WORD 202		
25801	112174	012706			.WORD ASYNC		
25802	112176	026302			.WORD DUMPER		
25803	112200	012746	010630	12\$:	MOV #WRD.10,-(SP)	:	
25804	112204	012746	011714		MOV #PHR.5,-(SP)	:	1111
25805	112210	012746	011524		MOV #WRD.72,-(SP)	:	
25806	112214	012746	010256		MOV #THR.FMT,-(SP)	:	
25807	112220	012746	000004		MOV #4,-(SP)	:	
25808	112224	010600			MOV SP,R0	: SP,*	
25809	112226	104414			TRAP 14	:	
25810	112230	005016			CLR (SP)	:	1111
25811	112232	005777	103226		TST @ML.REG+60	:	
25812	112236	100001			BPL 13\$:	
25813	112240	005216			INC (SP)	:	
25814	112242	010246		13\$:	MOV R2,-(SP)	: EXPECTED,*	
25815	112244	012746	006442		MOV #FMT.1,-(SP)	:	
25816	112250	012746	000003		MOV #3,-(SP)	:	
25817	112254	010600			MOV SP,R0	: SP,*	
25818	112256	104414			TRAP 14	:	
25819	112260	062706	000020		ADD #20,SP	:	1110
25820	112264	104467		14\$:	TRAP 67	:	1111
25821	112266	006000			ROR R0	:	
25822	112270	103634			BLO 2\$:	
25823	112272	005201			INC R1	: LOOP	1107
25824	112274	020127	000002		CMP R1,#2	: LOOP,*	
25825	112300	003630			BLE 2\$:	
25826	112302	022626			CMP (SP)+,(SP)+	:	1102
25827	112304	000207			RTS PC	:	
25828							
25829					: Routine Size: 141 words		
25830					: Maximum stack depth per invocation: 14 words		
25835							
25836							
25840							
25841					.SBTTL T59 TEST CODE SECTION		

25846
 25847
 25848
 25849 112306
 25850 112306 004767 177342
 25851 112312 104466
 25852 112314 006000
 25853 112316 103773
 25854 112320 000207
 25855
 25856
 25857
 25862
 25863
 25864 ; 11122 !<BLF/PAGE>

;ML4AD
 ;
 TEST CODE SECTION

29-Mar-1982 16:23:04 TOPS
 29-Mar-1982 16:21:03 PA:<

T59::
 1\$: JSR PC,\$T59 ;
 TRAP 66
 ROR R0
 BLO 1\$
 RTS PC

1111

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

25866 :ML4AD

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (112)

```

25867 : TEST CODE SECTION
25868 :
25869 : 11123 !
25870 : 11124 BGNTST:
25871 : 11125
25872 : 11126 !++
25873 : 11127 TEST NUMBER: TST 60
25874 : 11128
25875 : 11129 TEST NAME: PROM DATA TEST
25876 : 11130
25877 : 11131 TEST DESCRIPTION:
25878 : 11132
25879 : 11133 VERIFY THAT CHECK SUM VALUES FOR
25880 : 11134 ALL PRESENT MEMORY ARRAY UV PROM
25881 : 11135 LOCATIONS ARE ERROR FREE BY:
25882 : 11136
25883 : 11137 1. DOING MBUS TRANSFERS
25884 : 11138 AT ALL PRESENT BLOCKS
25885 : 11139 AND TESTING THE UNS BIT
25886 : 11140 TO BE CLEARED.
25887 : 11141
25888 : 11142 IMPLICIT INPUTS:
25889 : 11143
25890 : 11144 IO_BUF
25891 : 11145
25892 : 11146 A VECTOR OF 256 WORDS WHERE
25893 : 11147 DATA FOR MBUS READ AND WRITE
25894 : 11148 FUNCTIONS IS FOUND.
25895 : 11149
25896 : 11150
25897 : 11151 CHIP SIZ
25898 : 11152 INDICATED THE SIZE OF THE
25899 : 11153 ARRAY MODULES MOS RAMS.
25900 : 11154
25901 : 11155
25902 : 11156 !--
25903 : 11157
25904 : 11158 CLR_THRESHOLD;
25905 : 11159
25906 : 11160 !
25907 : 11161 !VERSION CZMLAD CHANGED TEST TO UNSIGNED TEST
25908 : 11162 !
25909 : 11163 incrU DSA_CNT from 0 to .LST_BLK do
25910 : 11164 begin
25911 : 11165 BGNSUB;
25912 : 11166 CLR_MBUS;
25913 : 11167 ECC_DIS = ONE;
25914 : 11168 MLWC = not 255;
25915 : 11169 MLBA = IO_BUF;
25916 : 11170 MLDA = .DSA_CNT;
25917 : 11171 MLCS1 = write;
25918 : 11172 TIME_OUT_LOOP;
25919 : 11173
25920 : 11174 if .UNS IS_SET

```

!CLEAR ERROR PRINT THRESHOLD

!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

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD BLI.4 (112)

```

25922 :ML4AD
25923 :
25924 :
25925 : 11175 then
25926 : 11176 begin
25927 : 11177 CMP THRESHOLD: !ERROR IF SET
25928 : 11178 ERRDF (107, ARR_DAT, DUMPER): !COMPARE ERROR PRINT THRESHOLD
25929 : 11179 PRINTB (TWO_FMT, WRD_35, PHR_4);
25930 : 11180 PRINTB (FMT_9, .DSA_CNT);
25931 : 11181
25932 : 11182 if .CHIP_SIZ eql 64 !NEED TO KNOW CHIP SIZE TO PRINT ARRAY NUMBER
25933 : 11183 then
25934 : 11184 begin
25935 : 11185 PRINTB (FMT_8, ((.DSA_CNT<11, 4>) + 1)); !64K MOS RAM
25936 : 11186 end
25937 : 11187 else
25938 : 11188 begin
25939 : 11189 PRINTB (FMT_8, ((.DSA_CNT<9, 4>) + 1)); !16K MOS RAM
25940 : 11190 end
25941 : 11191
25942 : 11192 end;
25943 : 11193
25944 : 11194 ENDSUB;
25945 : 11195 end;
25946 : 11196
25947 : 11197 ENDTST;

```

25951									
25952					.SBTTL	\$T60 TEST CODE SECTION			
25956	112322	004167	072212	\$T60:	JSR	R1, \$SAVE2	:		1112
25957	112326	005067	103044		CLR	P.CNT	:		1112
25958	112332	016702	101456		MOV	LST.BLK, R2	:		1116
25959	112336	005001			CLR	R1	:	DSA.CNT	
25960	112340	000544			BR	7\$:		
25961	112342	104402		1\$:	TRAP	2	:		1116
25962	112344	152777	000040	103072	BISB	#40, @ML.REG+40	:		1116
25963	112352	016700	103454		MOV	ML.DUT, R0	:		
25964	112356	042700	177770		BIC	#177770, R0	:		
25965	112362	142777	000007	103054	BICB	#7, @ML.REG+40	:		
25966	112370	150077	103050		BISB	R0, @ML.REG+40	:		
25967	112374	152777	000002	103122	BISB	#2, @ML.REG+120	:		1116
25968	112402	012777	177400	103004	MOV	#-400, @ML.REG+10	:		1116
25969	112410	012777	014022	103006	MOV	#10.BUF, @ML.REG+20	:		1116
25970	112416	010177	103012		MOV	R1, @ML.REG+30	:	DSA.CNT, *	1117
25971	112422	012777	000061	102754	MOV	#61, @ML.REG	:		1117
25972	112430	105777	103020	2\$:	TSTB	@ML.REG+50	:		
25973	112434	100375			BPL	2\$:		
25974	112436	032777	040000	103020	BIT	#40000, @ML.REG+60	:		1117
25975	112444	001476			BEQ	5\$:		
25976	112446	005267	102724		INC	P.CNT	:		1117

```

25978          :ML4AD
25979          :
25980          : TEST CODE SECTION
25981 112452 026767 102720 102720      CMP      P.CNT,LIMIT
25982 112460 003073                    BGT      6$
25983 112462 104455                    TRAP    55
25984 112464 000153                    .WORD   153
25985 112466 013012                    .WORD   ARR.DAT
25986 112470 026302                    .WORD   DUMPER
25987 112472 012746 011676            MOV      #PHR.4, -(SP)
25988 112476 012746 011110            MOV      #WRD.35, -(SP)
25989 112502 012746 010246            MOV      #TWO.FMT, -(SP)
25990 112506 012746 000003            MOV      #3, -(SP)
25991 112512 010600                    MOV      SP,R0
25992 112514 104414                    TRAP    14
25993 112516 010116                    MOV      R1, (SP)
25994 112520 012746 007064            MOV      #FMT.9, -(SP)
25995 112524 012746 000002            MOV      #2, -(SP)
25996 112530 010600                    MOV      SP,R0
25997 112532 104414                    TRAP    14
25998 112534 026727 101252 000100     CMP      CHIP.SIZ,#100
25999 112542 001020                    BNE     3$
26000 112544 010100                    MOV      R1,R0
26001 112546 006200                    ASR     R0
26002 112550 006200                    ASR     R0
26003 112552 006200                    ASR     R0
26004 112554 000300                    SWAB   R0
26005 112556 042700 177760            BIC     #177760,R0
26006 112562 010046                    MOV     R0, -(SP)
26007 112564 005216                    INC     (SP)
26008 112566 012746 007030            MOV     #FMT.8, -(SP)
26009 112572 012746 000002            MOV     #2, -(SP)
26010 112576 010600                    MOV     SP,R0
26011 112600 104414                    TRAP   14
26012 112602 000415                    BR     4$
26013 112604 010100                    3$: MOV   R1,R0
26014 112606 006200                    ASR    R0
26015 112610 000300                    SWAB  R0
26016 112612 042700 177760            BIC   #177760,R0
26017 112616 010046                    MOV   R0, -(SP)
26018 112620 005216                    INC   (SP)
26019 112622 012746 007030            MOV   #FMT.8, -(SP)
26020 112626 012746 000002            MOV   #2, -(SP)
26021 112632 010600                    MOV   SP,R0
26022 112634 104414                    TRAP  14
26023 112636 062706 000022            4$: ADD  #22,SP
26024 112642 104467                    5$: TRAP 67
26025 112644 006000                    ROR   R0
26026 112646 103635                    BLO  1$
26027 112650 005201                    6$: INC  R1
26028 112652 020102                    7$: CMP  R1,R2
26029 112654 101632                    BLOS  1$
26030 112656 000207                    RTS   PC
26031
26032

```

: Routine Size: 111 words

26034
26035
26036
26037
26042
26043
26047
26048
26052 112660
26053 112660 004767 177436
26054 112664 104466
26055 112666 006000
26056 112670 103773
26057 112672 000207
26058
26059
26060
26065
26066
26067 : 11198 !<BLF/PAGE>

:ML4AD
:
TEST CODE SECTION
: Maximum stack depth per invocation: 12 words

.SBTTL T60 TEST CODE SECTION
T60::
1\$: JSR PC,\$T60 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC

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

1119

29-Mar-1982 16:23:04
29-Mar-1982 16:21:03

TOPS-20 Bliss-16 V2(212)
PA:<NEALE>ML4AD.BLI.4 (113)

```

26069 :ML4AD
26070 :
26071 :
26072 :
26073 :
26074 :
26075 :
26076 :
26077 :
26078 :
26079 :
26080 :
26081 :
26082 :
26083 :
26084 :
26085 :
26086 :
26087 :
26088 :
26089 :
26090 :
26091 :
26092 :
26093 :
26094 :
26095 :
26096 :
26097 :
26098 :
26099 :
26100 :
26101 :
26102 :
26103 :
26104 :
26105 :
26106 :
26107 :
26108 :
26109 :
26110 :
26111 :
26112 :
26113 :
26114 :
26115 :
26116 :
26117 :
26118 :
26119 :
26120 :
26121 :
26122 :
26123 :

```

```

TEST CODE SECTION
11199 :
11200 :
11201 BGNTST;
11202 :
11203 :
11204 :
11205 :
11206 :
11207 :
11208 :
11209 :
11210 :
11211 :
11212 :
11213 :
11214 :
11215 :
11216 :
11217 local
11218 DSA CNT,
11219 SIZE,
11220 DST,
11221 SRC,
11222 INIT_ERR,
11223 BB_ERR;
11224 :
11225 :
11226 :
11227 :
11228 :
11229 :
11230 :
11231 if .INTERVEN
11232 then
11233 begin
11234 :
11235 :
11236 :
11237 :
11238 :
11239 :
11240 :
11241 :
11242 :
11243 :
11244 :
11245 :
11246 :
11247 :
11248 :
11249 :
11250 :

```

```

!BEGIN THE TEST

TEST NUMBER: TST 61
TEST NAME: BATTERY BACK-UP TEST

TEST DESCRIPTION:
THIS TEST IS DESIGNED SPECIFICALLY FOR THE
MARLBORO MFG'ING VERIFY STATION. ITS PURPOSE
IS TO TEST THE ML-11 SYSTEM FOR BATTERY BACK-UP DURING
POWER LOSS AND NO ECC INITIALIZE DURING POWER
UP WITH GOOD BATTERY BACK-UP.

--

!COUNTS THE NUMBER OF BLOCKS TRANSFERED
!STORES THE SIZE OF MASS BUS TRANSFERS
!STORES THE DESTINATION ADRS OF MASS BUS TRANSFERS
!STORES THE SOURCE ADRS OF MASS BUS TRANSFERS
!COUNTS THE # OF WORDS FOUND TO HAVE ZEROES DATA
!COUNTS THE # OF WORDS FOUND TO HAVE NONE ONES DATA BUT NOT 0'S DATA

PERFORM THIS TEST ONLY IF THE OPERATOR HAS
ENABLED MANUAL INTERVENTION TEST VIA THE S/W
QUESTIONS.

!DO THIS TEST IF INTERVEN IS SET

PRINT WHICH TEST NUMBER THIS

PRINTB (ONE_FMT, T_61):

TO START THE TEST OFF WRITE THE ENTIRE ML-11 MEMORY
WITH ALL ONES DATA. THIS WILL BE THE BACK GROUND FOR
THE ECC INITIALIZE PORTION OF THIS TEST. THIS ALL ONES
DATA PATTERN SHOULD STILL BE IN MEMORY AFTER THE POWER
IS TURNED BACK ON WITH GOOD BATTERY BACK-UP.

CLR MBUS;
IO_BUF = ONES;

!CLEAR THE MASS BUS
!LOAD THE FIRST IO BUF WORD WITH ONES DATA

```


26125	:ML4AD		29-Mar-1982 16:23:04	TOPS-20 BLISS-16 V2(212)
26126	:	TEST CODE SECTION	29-Mar-1982 16:21:03	PA:<NEALE>ML4AD.BLI.4 (113)
26127	:			
26128	:	11251 DSA_CNT = -1;		!RESET THE DSA COUNTER
26129	:	11252		
26130	:	11253 do		!LOAD ALL OF MEMORY WITH ONE'S DATA
26131	:	11254 begin		
26132	:	11255 BREAK;		!LOOK FOR CONTROL C
26133	:	11256 DSA_CNT = .DSA_CNT + 1;		!UP THE DSA COUNTER
26134	:	11257 WRT_TRANSFER (SIZE = -256, DST = .DSA_CNT, SRC = IO_BUF);		!DO THE WRITE XFER
26135	:	11258 end		
26136	:	11259		
26137	:	11260 !VERSION CZMLAD CHANGED TEST TO UNSIGNED TEST		
26138	:	11261		
26139	:	11262 until .DSA_CNT eqU .LST_BLK;		!REPEAT WRITING UNTIL THE LAST BLOCK IS DONE
26140	:	11263		
26141	:	11264		
26142	:	11265		
26143	:	11266 SET THE COLUMN VALID BIT 'VV'		
26144	:	11267 AND TELL THE OPERATOR TO TURN THE BOX		
26145	:	11268 POWER OFF WITH GOOD BATTERY BACK-UP.		
26146	:	11269 THE BACK GROUND PATTERN OF ALL ONES SHOULD		
26147	:	11270 BE MAINTAINED BY BATTERY BACK-UP AND ECC INIT		
26148	:	11271 ON POWER UP SHOULD NOT CLEAR THE PATTERN OUT.		
26149	:	11272		
26150	:	11273		
26151	:	11274 CLR MBUS;		!CLEAR THE MASS BUS
26152	:	11275 MLC51 = RD IN PRE;		!SET THE VV BIT via A RD IN PRE FUNC
26153	:	11276 PRINTB (ONE_FMT, PWR_OFF);		!TELL OPERATOR TO SHUT POWER OFF
26154	:	11277		
26155	:	11278 do		!DO NOTHING
26156	:	11279 begin		
26157	:	11280 BREAK;		!LOOK FOR CONTROL C
26158	:	11281 end		
26159	:	11282 until not (.MOL);		!UNTIL THE DRIVE POWER IS OFF
26160	:	11283		
26161	:	11284 DELAY (TWO_TH_US);		!DELAY FOR BATTERY BACK-UP MAINTAIN MEMORY
26162	:	11285		
26163	:	11286		
26164	:	11287 NOW TELL THE OPERATOR TO POWER UP THE BOX		
26165	:	11288 WITH GOOD BATTERY BACK-UP. THEN TEST THE		
26166	:	11289 VV BIT TO BE STILL SET AND THE BACK GROUND		
26167	:	11290 PATTERN OF ALL ONES TO STILL BE IN MEMORY.		
26168	:	11291		
26169	:	11292		
26170	:	11293 PRINTB (ONE_FMT, PUP_BB);		!TELL OPERATOR TO TURN BOX POWER BACK ON
26171	:	11294		
26172	:	11295 do		!DO NOTHING
26173	:	11296 begin		
26174	:	11297 BREAK;		!LOOK FOR CONTROL C
26175	:	11298 end		
26176	:	11299 until .DRY;		!UNTIL THE DRIVE IS READY
26177	:	11300		
26178	:	11301 DELAY (10);		!WAIT FOR THE DRIVE TO SETTLE
26179	:	11302		

```

26181 :ML4AD
26182 :
26183 :
26184 : 11303 if not (.VV)
26185 : 11304 then
26186 : 11305 begin
26187 : 11306 ERRDF (154, ASYNC, DUMPER);
26188 : 11307 PRINTB (ONE_FMT, BB_VV_ERR);
26189 : 11308 end;
26190 : 11309
26191 : 11310 INIT ERR = ZERO;
26192 : 11311 BB ERR = ZERO;
26193 : 11312 DSA_CNT = -1;
26194 : 11313
26195 : 11314 do
26196 : 11315 begin
26197 : 11316 BREAK;
26198 : 11317 DSA_CNT = .DSA_CNT + 1;
26199 : 11318 WRT_CHK_TRANSFER (SIZE = -256, DST = .DSA_CNT, SRC = IO_BUF);
26200 : 11319
26201 : 11320 if .WCE
26202 : 11321 then
26203 : 11322 begin
26204 : 11323
26205 : 11324 if .MLDB eql ZERO
26206 : 11325 then
26207 : 11326 INIT_ERR = .INIT_ERR + 1
26208 : 11327 else
26209 : 11328 BB_ERR = .BB_ERR + 1;
26210 : 11329
26211 : 11330 end;
26212 : 11331
26213 : 11332 end
26214 : 11333
26215 : 11334 VERSION CZMLAD CHANGED TEST TO UNSIGNED TEST
26216 : 11335
26217 : 11336 until .DSA_CNT eqlU .LST_BLK;
26218 : 11337
26219 : 11338
26220 : 11339 IF THERE ARE MORE THAN 127 SECTORS IN
26221 : 11340 ERROR WITH ALL ZEROES DATA THEN THE
26222 : 11341 PROBLEM IS PROBABLY IN
26223 : 11342 THE ECC INITIALIZE LOGIC AND NOT IN THE
26224 : 11343 BATTERY BACK-UP OF MEMORY.
26225 : 11344
26226 : 11345 IF INIT_ERR IS NOT > 127 AND THERE ARE
26227 : 11346 > 100 WORDS READ WITH DATA OTHER THAN
26228 : 11347 ALL ONES OR ALL ZEROES THEN THE PROBLEM
26229 : 11348 IS PROBABLY CAUSED BY THE BATTERY BACK-UP
26230 : 11349 LOGIC NOT MAINTAINING THE MEMORY DURING
26231 : 11350 POWER FAILURE.
26232 : 11351
26233 : 11352
26234 : 11353 if .INIT_ERR gtr 127
26235 : 11354 then

```

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (113)

!DID POWERING UP WITH GOOD BB CLEAR THE VV BIT

!THE VV BIT GOT CLEARED
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS

!CLEAR THE INIT COUNTER
!CLEAR THE BATTERY BACK-UP COUNTER
!RESET THE DSA COUNTER

!WRITE CHECK MEMORY AND TEST FOR BAT BACK-UP AND/OR NO ECC INIT

!LOOK FOR CONTROL C
!UP THE DSA COUNTER
!DO THE WRITE CHECK XFER

!IS THIS BLOCK IN ERROR

!THIS BLOCK IS IN ERROR

!SEE WHAT KIND OF AN ERROR IT IS

!IT'S A INITIALIZE ERROR

!IT'S A BAT BACK-UP ERROR

!REPEAT WRITE CHECKING UNTIL THE LAST BLOCK

!IS THE ECC INIT LOGIC IN ERROR

29-Mar-1982 16:23:04 TOPS-20 Bliss-16 V2(212)
29-Mar-1982 16:21:03 PA:<NEALE>ML4AD.BLI.4 (113)

```

26237 :ML4AD
26238 :
26239 :
26240 :      11355      begin
26241 :      11356      ERRDF (155, ASYNC, DUMPER);
26242 :      11357      PRINTB (ONE_FMT, BB_INIT_ERR);
26243 :      11358      end
26244 :      11359      else
26245 :      11360      begin
26246 :      11361
26247 :      11362      if .BB_ERR gtr 100
26248 :      11363      then
26249 :      11364      begin
26250 :      11365      ERRDF (156, ASYNC, DUMPER);
26251 :      11366      PRINTB (ONE_FMT, BB_BB_ERR);
26252 :      11367      end;
26253 :      11368
26254 :      11369      end;
26255 :      11370
26256 :      11371      end;
26257 :      11372
26258 :      11373      ENDTST;

```

```

!THE ECC INIT LOGIC IS IN ERROR
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS

!THE ECC INIT LOGIC IS NOT IN ERROR

!IS THE BATTERY BACK-UP LOGIC IN ERROR

!THE BATTERY BACK-UP LOGIC IS IN ERROR
!REPORT THE ERROR TO THE OPERATOR
!TELL WHAT THE ERROR IS

!END THE TEST

```

Address	Hex	Hex	Hex	Label	Instruction	Comment	Address
26263				.SBTTL	\$T61 TEST CODE SECTION		
26267	112674	004167	071712	\$T61:	JSR R1,\$SAVES		1119
26268	112700	024646			CMP -(SP),-(SP)		
26269	112702	032767	000001 067500		BIT #1,INTERVEN		1123
26270	112710	001002			BNE 1\$		
26271	112712	000167	000600		JMP 20\$		
26272	112716	012746	005012	1\$:	MOV #T.61,-(SP)		1123
26273	112722	012746	010240		MOV #ONE_FMT,-(SP)		
26274	112726	012746	000002		MOV #2,-(SP)		
26275	112732	010600			MOV SP,R0	: SP,*	
26276	112734	104414			TRAP 14		
26277	112736	152777	000040 102500		BISB #40,2ML.REG+40		
26278	112744	016705	103062		MOV ML,DUT,R5		
26279	112750	042705	177770		BIC #177770,R5		
26280	112754	142777	000007 102462		BICB #7,2ML.REG+40		
26281	112762	150577	102456		BISB R5,2ML.REG+40		
26282	112766	012767	177777 101026		MOV #-1,IO.BUF		1125
26283	112774	012702	177777		MOV #-1,R2	*.DSA.CNT	1125
26284	113000	104422		2\$:	TRAP 22		1125
26285	113002	005202			INC R2	DSA.CNT	1125
26286	113004	012705	177400		MOV #-400,R5	*.SIZE	1125
26287	113010	010546			MOV R5,-(SP)	SIZE,*	1125
26288	113012	010266	000010		MOV R2,10(SP)	DSA.CNT,DST	
26289	113016	010246			MOV R2,-(SP)	DSA.CNT,*	
26290	113020	012701	014022		MOV #10,BUF,R1	*.SRC	
26291	113024	010146			MOV R1,-(SP)	SRC,*	

Address	Label	OpCode	Operand	Comment	Line
26349	:ML4AD				
26350	:			TEST CODE SECTION	
26351					
26352		MOV	#2,-(SP)		
26353		MOV	SP,R0	: SP,*	
26354		TRAP	14		
26355		ADD	#6,SP		
26356	13\$:	CLR	R3	: INIT.ERR	1130
26357		CLR	R4	: BB.ERR	1131
26358		MOV	#-1,R2	: *,DSA.CNT	1131
26359	14\$:	TRAP	22		1131
26360		INC	R2	: DSA.CNT	1131
26361		MOV	#-400,R5	: *,SIZE	1131
26362		MOV	R5,-(SP)	: SIZE,*	
26363		MOV	R2,20(SP)	: DSA.CNT,DST	
26364		MOV	R2,-(SP)	: DSA.CNT,*	
26365		MOV	#10.BUF,R1	: *,SRC	
26366		MOV	R1,-(SP)	: SRC,*	
26367		JSR	PC,WRT.CHK.TRANSFE		
26368		BIT	#4000,@ML.REG+40		
26369		BEQ	16\$		1132
26370		TST	@ML.REG+110		
26371		BNE	15\$		1132
26372		INC	R3	: INIT.ERR	1132
26373		BR	16\$		1132
26374	15\$:	INC	R4	: BB.ERR	1132
26375	16\$:	ADD	#6,SP		1132
26376		CMP	R2,LST.BLK	: DSA.CNT,*	1131
26377		BNE	14\$		1133
26378		CMP	R3,#177	: INIT.ERR,*	1135
26379		BLE	17\$		
26380		TRAP	55		1135
26381		.WORD	233		
26382		.WORD	ASYN		
26383		.WORD	DUMPER		
26384		MOV	#BB.INIT.ERR,-(SP)		1135
26385		MOV	#ONE.FMT,-(SP)		
26386		MOV	#2,-(SP)		
26387		MOV	SP,R0	: SP,*	
26388		TRAP	14		
26389		BR	18\$		1135
26390	17\$:	CMP	R4,#144	: BB.ERR,*	1136
26391		BLE	19\$		
26392		TRAP	55		1136
26393		.WORD	234		
26394		.WORD	ASYN		
26395		.WORD	DUMPER		
26396		MOV	#BB.BB.ERR,-(SP)		1136
26397		MOV	#ONE.FMT,-(SP)		
26398		MOV	#2,-(SP)		
26399		MOV	SP,R0	: SP,*	
26400		TRAP	14		
26401	18\$:	ADD	#6,SP		1136
26402	19\$:	ADD	#16,SP		1123
26403	20\$:	CMP	(SP)+,(SP)+		1119

26405
26406
26407
26408 113520 000207
26409
26410
26411
26416
26417
26421
26422
26426 113522
26427 113522 004767 177146
26428 113526 104466
26429 113530 006000
26430 113532 103773
26431 113534 000207
26432
26433
26434
26439
26440
26441 ;

:ML4AD
:
TEST CODE SECTION
RTS PC
: Routine Size: 203 words
: Maximum stack depth per invocation: 18 words

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

.SBTTL T61 TEST CODE SECTION
T61::
1\$: JSR PC,\$T61 ;
TRAP 66
ROR R0
BLO 1\$
RTS PC
: Routine Size: 6 words
: Maximum stack depth per invocation: 0 words

1137

11374 !<BLF/PAGE>

26443 :ML4AD

29-Mar-1982 16:23:04

TOPS-20 Bliss-16 V2(212)

29-Mar-1982 16:21:03

PA:<NEALE>ML4AD.BLI.4 (114)

TEST CODE SECTION

26444 :
26445 :
26446 : 11375 !
26447 : 11376 !
26448 : 11377 !
26449 : 11378 !
26450 : 11379 !
26451 : 11380 !
26452 : 11381 !
26453 : 11382 !
26454 : 11383 !
26455 : 11384 !
26456 : 11385 !
26457 : 11386 !
26458 : 11387 !
26459 : 11388 !
26460 : 11389 !
26461 : 11390 !
26462 : 11391 !
26463 : 11392 !
26464 : 11393 !
26465 : 11394 !++
26466 : 11395 BGNCLN;
26467 : 11396 CLR = ONE;
26468 : 11397 return;
26469 : 11398 ENDCLN;

CLEAN UP SECTION

THE CLEAN UP SECTION IS ENTERED WHEN:

1. A NORMAL END OF PASS IS DONE
ie. WHEN ONE COMPLETE PASS OF
THE DIAG IS DONE.
2. WHEN A DRIVE ERROR IS DETECTED AND
AND THE UNIT IS DROPPED AND THE DOCLN
MACRO IS CALLED.
3. WHEN A ^C IS TYPE AT THE CONSOLE TERMINAL
DURING EXECUTION OF A TEST.

IN ANY EVENT OF ENTERING THIS SECTION THE
ML11 CONTROL BOARD TESTER'S POWER SUPPLIES
ARE POWERED DOWN via THE ROUTINE CALL: ''

!BEGIN THE CLEAN UP CODE
!CLEAR THE MASS BUS
!RETURN TO DRS>
!END THE CLEAN UP CODE

26473 :
26474 :
26478 113536 152777 000040 101700 LCLEAN: .SBTTL LCLEAN TEST CODE SECTION
26479 113544 000207 BLSB #40,AML.REG+40 : 1139
RTS PC : 1137
26480 :
26481 : Routine Size: 4 words
26482 : Maximum stack depth per invocation: 0 words
26487 :
26488 :
26492 :
26493 :
26497 113546 LSCLEAN: .SBTTL LSCLEAN TEST CODE SECTION

```
26499 ;ML4AD
26500 ;
26501 ; TEST CODE SECTION
26502 113546 004767 177764 JSR PC,LCLEAN
26503 113552 104412 TRAP 12
26504 113554 000207 RTS PC
26505
26506 ; Routine Size: 4 words
26507 ; Maximum stack depth per invocation: 0 words
26512
26513
26514 : 11399 LASTAD; !DEFINE THE DIAG'S LAST ADDRESS
26515 : 11400 BGNSETUP (0); !SET UP ZERO P-TABLES
26516 : 11401 ENDSETUP; !END THE SETUP
26520
26521
26522 113556 113562 BLSLAS::.WORD TSFREE
26523 113560 000000 .WORD <<TSFREE-<BLSLAS+4>>/2>
26524 113562 000000 TSFREE::.WORD 0
26525
26526
26527 113562 L$LAST== BLSLAS+4
26528 000000 T$PTHV== 0
26529
26530
26531 .SBTTL SEND.LINK TEST CODE SECTION
26535 113564 SEND.LINK::
26536 113564 000207 RTS PC
26537 ;
26538 ; Routine Size: 1 word
26539 ; Maximum stack depth per invocation: 0 words
26544
26545
26546 : 11402 end
26547 : 11403
26548 : 11404 eludom
26552
```

29-Mar-1982 16:23:04 TOPS
29-Mar-1982 16:21:03 PA:<

1139

1139

26554
26555
26556
26557
26558
26559
26560
26561
26562
26563
26564
26565
26566
26567
26568
26569
26570
26571
26572
26573

:ML4AD
:
: OTS external references
 .GLOBL BL\$GT2, \$\$SAVE5, \$\$SAVE4, \$\$SAVE3
 .GLOBL \$\$SAVE2, BL\$PU2, BL\$PU1, BL\$GT1
 .GLOBL BL\$DIV, BL\$MOD, BL\$MUL

: Size: 15786 code + 2346 data words
: Run Time: 02:46.0
: Elapsed Time: 03:20.7
: Memory Used: 124 pages
: Compilation Complete

000001

.END

AJR = 000020 G	C\$CLCK= 000062	D2.TEM= 013376	FNC.19= 012426	HELP = 000000
ARR.DA= 013012	C\$CLEA= 000012	EF.COM= 000036 G	FNC.2 = 012164	HOE = 100000 G
ARR.IN 014004	C\$CLOS= 000035	EF.NEW= 000035 G G	FNC.21= 012440	HW.OR. 013402
ARR.16 014016	C\$CLP1= 000006	EF.PWR= 000034 G G	FNC.22= 012452	IBE = 010000 G
ASSEMB= 000010	C\$CYEC= 000036	EF.RES= 000037 G G	FNC.23= 012464	IDU = 000040 G G
ASYN = 012706	C\$DCLN= 000044	EF.STA= 000040 G	FNC.3 = 012172	IER = 020000 G
A.CAL 015362	C\$DODU= 000051	EIG.FM= 010364	FNC.4 = 012200	INTER = 013114
A.GEN 015370	C\$DRPT= 000024	ELV.FM= 010466	FNC.5 = 012216	INTERV 002410 G
BB.BB. = 006334	C\$DU = 000053	ERRBLK 002160 G	FNC.6 = 012226	IO.BUF 014022
BB.INI= 006242	C\$EDIT= 000003	ERRMSG 002156 G G	FNC.7 = 012234	ISR = 000100 G G
BB.VV. = 006150	C\$ERDF= 000055	ERRNBR 002154 G G	FNC.8 = 012244	IXE = 004000 G
BIT0 = 000001 G	C\$ERHR= 000056	ERRTHR 002402 G G	FNC.9 = 012260	ISAU = 000041
BIT00 = 000001 G G	C\$ERRO= 000060	ERRTYP 002152 G	FORCE. 020032	ISAUTO= 000041
BIT01 = 000002 G	C\$ERSF= 000054	ERR.CH 017442	FOR.EC 020210	ISCLN = 000041
BIT02 = 000004 G	C\$ERSO= 000057	EVL = 000004 G	FOR.FM= 010270	ISDU = 000041
BIT03 = 000010 G	C\$ESCA= 000010	ESEND = 002100	FSAU = 000015	ISHRD = 000041
BIT04 = 000020 G	C\$ESEG= 000005	ESLOAD= 000035	FSAUTO= 000020	ISINIT= 000041
BIT05 = 000040 G	C\$ESUB= 000003	E2.TEM= 013400	FSBGN = 000040	ISMOD = 000041
BIT06 = 000100 G	C\$ETST= 000001	FIND.C 017660	FSCLEA= 000007	ISMSG = 000041
BIT07 = 000200 G	C\$EXIT= 000032	FIRST. 020262	FSDU = 000016	ISPROT= 000040
BIT08 = 000400 G	C\$GETB= 000026	FIV.FM= 010304	FSEND = 000041	ISPTAB= 000041
BIT09 = 001000 G	C\$GETW= 000027	FMT.1 = 006442	FSHARD= 000004	ISPWR = 000041
BIT1 = 000002 G	C\$GMAN= 000043	FMT.10= 007116	FSHW = 000013	ISRPT = 000041
BIT10 = 002000 G	C\$GPHR= 000042	FMT.11= 007170	FSINIT= 000006	ISSEG = 000041
BIT11 = 004000 G	C\$GPLO= 000030	FMT.12= 007224	FSJMP = 000050	ISSETU= 000041
BIT12 = 010000 G	C\$GPRI= 000040	FMT.13= 007254	FSMGD = 000000	ISSFT = 000041
BIT13 = 020000 G	C\$INIT= 000011	FMT.14= 007330	FSMSG = 000011	ISSRV = 000041
BIT14 = 040000 G	C\$INLP= 000020	FMT.15= 007370	FSPROT= 000021	ISSUB = 000041
BIT15 = 100000 G	C\$MANI= 000050	FMT.16= 007436	FSPWR = 000017	ISTST = 000041
BIT2 = 000004 G	C\$MEM = 000031	FMT.17= 007526	FSRPT = 000012	JSJMP = 000167
BIT3 = 000010 G	C\$MSG = 000023	FMT.18= 007562	FSSEG = 000003	LAST.B 020344
BIT4 = 000020 G	C\$OPEN= 000034	FMT.19= 007632	FSSOFT= 000005	LAU 004704
BIT5 = 000040 G	C\$PNTB= 000014	FMT.2 = 006506	FSSRV = 000010	LAUTO 004660
BIT6 = 000100 G	C\$PNTF= 000017	FMT.20= 007702	FSSUB = 000002	LCLEAN 113536
BIT7 = 000200 G	C\$PNTS= 000016	FMT.21= 007754	FSSW = 000014	LDU 004672
BIT8 = 000400 G	C\$PNTX= 000015	FMT.22= 010026	FSTEST= 000001	LD.LNG 021354
BIT9 = 001000 G	C\$QIO = 000377	FMT.23= 010072	GD.BLK 020312	LIMIT 015400
BL\$DIV 004446 G	C\$RDBU= 000007	FMT.24= 010142	GOOD.B 014006	LINIT 027650
BL\$GT1 003512 G	C\$REFG= 000047	FMT.25= 010166	G\$CNT0= 000200	LOAD.S 016034
BL\$GT2 003634 G	C\$RESE= 000033	FMT.26= 010206	G\$DELM= 000372	LOE = 040000 G
BL\$LAS 113556 G	C\$REVI= 000003	FMT.3 = 006574	G\$DISP= 000003	LOT = 000010 G
BL\$MOD 004460 G	C\$RFLA= 000021	FMT.4 = 006620	G\$EXCP= 000400	LRPT 004646
BL\$MUL 004222 G	C\$RPT = 000025	FMT.5 = 006650	G\$HILI= 000002	LST.AR 014020
BL\$PU1 003776 G	C\$SEFG= 000046	FMT.6 = 006752	G\$LOLI= 000001	LST.BL 014014
BL\$PU2 004072 G	C\$SPRI= 000041	FMT.7 = 007002	G\$NO = 000000	LST.DU 015356
BL\$SHF 004472 G	C\$SVEC= 000037	FMT.8 = 007030	G\$OFFS= 000400	LSACP 002110 G
BOE = 000400 G	C\$TPRI= 000013	FMT.9 = 007064	G\$OFSI= 000376	LSAPT 002036 G G
B.CAL 015364	DATA.L= 013154	FNC.1 = 012150	G\$PRMA= 000001	LSAU 004706 G
B.GEN 015372	DAT.DM 016412	FNC.10= 012272	G\$PRMD= 000002	LSAUT 002070 G
CAL.CR 017042	DFPTBL 002360 G	FNC.11= 012302	G\$PRML= 000000	LSAUTO 004662 G
CHIP.S 014012	DIAGMC= 000000	FNC.12= 012322	G\$RADA= 000140	LSCCP 002106 G
C\$AU = 000052	DIVMOD 004264	FNC.13= 012334	G\$RADB= 000000	LSCLEA 113546 G
C\$AUTO= 000061	DRIVE. 015354	FNC.14= 012344	G\$RADD= 000040	LSCO 002032 G
C\$BRK = 000022	DROP.C 015402	FNC.15= 012360	G\$RADL= 000120	LSDEPO 002011 G
C\$BSEG= 000004	DT.1 015764	FNC.16= 012372	G\$RADO= 000020	LSDESC 002130 G
C\$BSUB= 000002	DUMPER 026302 G	FNC.17= 012404	G\$XFER= 000004	LSDESP 002076 G
C\$CEFG= 000045	D1.TEM= 013374	FNC.18= 012414	G\$YES = 000010	LSDEVP 002060 G

LSDISP	002164	G	MSGH7	003052	P.AAC	005112	P.ACH	010566	P.AEM	011454
LSDLY	002116	G	MSGH1	003160	P.AAD	005202	P.ACI	010602	P.AEN	011462
LSDTP	002040	G	MSGH2	003224	P.AAE	005272	P.ACJ	010616	P.AEO	011470
LSDTYP	002034	G	MSGH3	003270	P.AAF	005340	P.ACK	010630	P.AEP	011474
LSDU	004674	G	MSGH4	003334	P.AAG	005414	P.ACL	010640	P.AEQ	011502
LSDUT	002072	G	MSGH5	003400	P.AAH	005470	P.ACM	010650	P.AER	011514
LSDVTY	002122	G	MSDUMP	026312	P.AAI	005570	P.ACN	010660	P.AES	011524
LSEF	002052	G	NIB.SA	013374	P.AAJ	005652	P.ACO	010664	P.AET	011532
LSENV1	002044	G	NIN.FM=	010410	P.AAK	005722	P.ACP	010676	P.AEU	011540
LSERRT	002152	G	NO.INI=	005470	P.AAL	005764	P.ACQ	010704	P.AEV	011552
LSETP	002102	G	ONEFIL=	000001	P.AAM	006060	P.ACR	010712	P.AEW	011562
LSEXF1	002046	G	ONEPAS	002406	P.AAN	006150	P.ACS	010724	P.AEX	011570
LSEXP4	002064	G	ONE.FM=	010240	P.AAO	006242	P.ACT	010730	P.AEY	011610
LSEXP5	002066	G	OP.NUM	014002	P.AAP	006334	P.ACU	010736	P.AEZ	011626
LSHARD	002414	G	OSAPTS=	000001	P.AAQ	006442	P.ACV	010744	P.AFA	011644
LSHIME	002120	G	OSAU =	000001	P.AAR	006506	P.ACW	010752	P.AFB	011676
LSHPCP	002016	G	OSBGNR=	000001	P.AAS	006574	P.ACX	010766	P.AFC	011714
LSHPTP	002022	G	OSBGNS=	000001	P.AAT	006620	P.ACY	010774	P.AFD	011726
LSHW	002360	G	OSDU =	000001	P.AAU	006650	P.ACZ	011002	P.AFE	011740
LSICP	002104	G	OSERRT=	000001	P.AAV	006752	P.ADA	011016	P.AFF	011762
LSINIT	030714	G	OSGNSW=	000001	P.AAW	007002	P.ADB	011024	P.AFG	011774
LSLADP	002026	G	OSPOIN=	000001	P.AAX	007030	P.ADC	011032	P.AFH	012006
LSLAST=	113562	G	OSSETU=	000001	P.AAY	007064	P.ADD	011046	P.AFI	012020
LSLOAD	002100	G	PAR.DI	014010	P.AAZ	007116	P.ADE	011054	P.AFJ	012036
LSLUN	002074	G	PD.TEM	015342	P.ABA	007170	P.ADF	011070	P.AFK	012060
LSMREV	002050	G	PHR.1 =	011610	P.ABB	007224	P.ADG	011076	P.AFL	012104
LSNAME	002000	G	PHR.10=	012006	P.ABC	007254	P.ADH	011102	P.AFM	012136
LSPRIO	002042	G	PHR.11=	012020	P.ABD	007330	P.ADI	011110	P.AFN	012150
LSPROT	003444	G	PHR.12=	012036	P.ABE	007370	P.ADJ	011116	P.AFO	012164
LSPRT	002112	G	PHR.13=	012060	P.ABF	007436	P.ADK	011122	P.AFP	012172
LSREPP	002062	G	PHR.14=	012104	P.ABG	007526	P.ADL	011132	P.AFQ	012200
LSREV	002010	G	PHR.15=	012136	P.ABH	007562	P.ADM	011140	P.AFR	012216
LSRPT	004650	G	PHR.2 =	011626	P.ABI	007632	P.ADN	011150	P.AFS	012226
LSSOFT	003122	G	PHR.3 =	011644	P.ABJ	007702	P.ADO	011154	P.AFT	012234
LSSPC	002056	G	PHR.4 =	011676	P.ABK	007754	P.ADP	011170	P.AFU	012244
LSSPCP	002020	G	PHR.5 =	011714	P.ABL	010026	P.ADQ	011176	P.AFV	012260
LSSPTP	002024	G	PHR.6 =	011726	P.ABM	010072	P.ADR	011206	P.AFW	012272
LSSTA	002030	G	PHR.7 =	011740	P.ABN	010142	P.ADS	011214	P.AFX	012302
LSSW	002400	G	PHR.8 =	011762	P.ABO	010166	P.ADT	011226	P.AFY	012322
LSTEST	002114	G	PHR.9 =	011774	P.ABP	010206	P.ADU	011240	P.AFZ	012334
LSTIML	002014	G	PNT =	001000	P.ABQ	010240	P.ADV	011250	P.AGA	012344
LSUNIT	002012	G	PRI =	002000	P.ABR	010246	P.ADW	011260	P.AGB	012360
L10000	002376	G	PRI00 =	000000	P.ABS	010256	P.ADX	011270	P.AGC	012372
L10001	002412	G	PRI01 =	000040	P.ABT	010270	P.ADY	011276	P.AGD	012404
L10002	002510	G	PRI02 =	000100	P.ABU	010304	P.ADZ	011310	P.AGE	012414
L10003	003160	G	PRI03 =	000140	P.ABV	010322	P.AEA	011316	P.AGF	012426
MB.DIS=	005570	G	PRI04 =	000200	P.ABW	010342	P.AEB	011326	P.AGG	012440
MEM.AR=	013052	G	PRI05 =	000240	P.ABX	010364	P.AEC	011336	P.AGH	012452
ML.DUT	016032	G	PRI06 =	000300	P.ABY	010410	P.AED	011344	P.AGI	012464
ML.LUN	016030	G	PRI07 =	000340	P.ABZ	010436	P.AEE	011352	P.AGJ	012500
ML.REG	015404	G	PRSN	002400	P.ACA	010466	P.AEF	011362	P.AGK	012510
MSGH1	002510	G	PTBL.P	014000	P.ACB	010520	P.AEG	011374	P.AGL	012516
MSGH2	002556	G	PUP.BB=	006060	P.ACC	010524	P.AEH	011400	P.AGM	012524
MSGH3	002624	G	PWR.OF=	005112	P.ACD	010536	P.AEI	011416	P.AGN	012532
MSGH4	002670	G	PWR.ON=	005202	P.ACE	010544	P.AEJ	011432	P.AGO	012540
MSGH5	002736	G	P.AAA	004716	P.ACF	010552	P.AEK	011440	P.AGP	012546
MSGH6	003004	G	P.AAB	005012	P.ACG	010560	P.AEL	011446	P.AGQ	012554

P.AGR	012562	REG.20=	012656	T\$TEMP=	000000	T51	103202	G	WRD.43=	011176
P.AGS	012570	REG.21=	012676	T\$TEST=	000000	T52	104020	G	WRD.44=	011206
P.AGT	012576	REG.3 =	012516	T\$TSTM=	177777	T53	104660	G	WRD.45=	011214
P.AGU	012604	REG.4 =	012524	T\$TSTS=	000000	T54	105722	G	WRD.46=	011226
P.AGV	012612	REG.5 =	012532	T\$SHAR=	010002	T55	106612	G	WRD.47=	011240
P.AGW	012620	REG.6 =	012540	T\$SHW =	010000	T56	107742	G	WRD.48=	011250
P.AGX	012626	REG.7 =	012546	T\$SPRO=	010004	T57	111234	G	WRD.49=	011260
P.AGY	012634	REG.8 =	012554	T\$SOF=	010003	T58	111640	G	WRD.5 =	010552
P.AGZ	012642	REG.9 =	012562	T\$SSW =	010001	T59	112306	G	WRD.50=	011270
P.AHA	012652	REM.TB	015664	T.21 =	004716	T6	033476	G	WRD.51=	011276
P.AHB	012660	RE2	004640	T.61 =	005012	T60	112660	G	WRD.52=	011310
P.AHC	012666	RE3	004636	T1	031134	T61	113522	G	WRD.53=	011316
P.AHD	012676	RE4	004634	T10	036266	T7	034354	G	WRD.54=	011326
P.AHE	012706	RH.ADD	016022	T11	036510	T8	035370	G	WRD.55=	011336
P.AHF	012750	RH.ERR=	013302	T12	037066	T9	035624	G	WRD.56=	011344
P.AHG	013012	RH.TYP	016024	T13	040230	UAM =	000200	G	WRD.57=	011352
P.AHH	013052	RH.VEC	016026	T14	041462	UNS.ER=	005272	WRD.58=	011362	
P.AHI	013114	SC.SET=	013214	T15	042714	VV.CLE=	005414	WRD.59=	011374	
P.AHJ	013154	SEV.FM=	010342	T16	043556	VV.NOT=	005340	WRD.6 =	010560	
P.AHK	013214	SFPTBL	002400	T17	045044	WC.SET=	005764	WRD.60=	011400	
P.AHL	013250	SIX.FM=	010322	T18	045260	WC.ERR=	005652	WRD.61=	011416	
P.AHM	013302	STACK	015034	T19	046430	WRD.1 =	010520	WRD.62=	011432	
P.AHN	013330	STK.OF	015022	T2	031460	WRD.10=	010630	WRD.63=	011440	
P.CAL	015366	STRIPP	016444	T20	047430	WRD.11=	010640	WRD.64=	011446	
P.CNT	015376	SVCGBL=	177777	T21	051176	WRD.12=	010650	WRD.65=	011454	
P.GEN	015374	SVCINS=	177777	T22	051724	WRD.13=	010660	WRD.67=	011462	
RAS.IN	015346	SVCSUB=	177777	T23	052262	WRD.14=	010664	WRD.68=	011470	
RD.CS1	021762	SVCTAG=	177777	T24	053456	WRD.15=	010676	WRD.69=	011474	
RD.DA	022316	SVCTST=	177777	T25	054270	WRD.16=	010704	WRD.7 =	010566	
RD.DAT	015352	SYNC =	012750	T26	055466	WRD.17=	010712	WRD.70=	011502	
RD.DS	025216	S\$LSYM=	010000	T27	056244	WRD.18=	010724	WRD.71=	011514	
RD.D1	024104	TEN.FM=	010436	T28	057230	WRD.19=	010730	WRD.72=	011524	
RD.D2	024416	THR.FM=	010256	T29	060342	WRD.2 =	010524	WRD.73=	011532	
RD.D3	024744	TIME.C=	013330	T3	031666	WRD.20=	010736	WRD.74=	011540	
RD.EE	023760	TRBLE. =	013250	T30	061136	WRD.21=	010744	WRD.75=	011552	
RD.EL	023716	TST.LN	020376	T31	062534	WRD.22=	010752	WRD.76=	011562	
RD.ER	022140	TWO.FM=	010246	T32	064272	WRD.23=	010766	WRD.77=	011570	
RD.E1	023074	T\$ARGC=	000002	T33	065304	WRD.24=	010774	WRD.8 =	010602	
RD.E2	023314	T\$CODE=	004130	T34	066266	WRD.25=	011002	WRD.9 =	010616	
RD.MR	022474	T\$ERRN=	000000	T35	066744	WRD.26=	011016	WRT.CH	027156	
RD.PA	022666	T\$EXCP=	000000	T36	070030	WRD.27=	011024	WRT.CS	021712	
RD.PD	023556	T\$FREE	113562	T37	070756	WRD.29=	011032	WRT.DA	022246	
RD.REG	025712	T\$GMAN=	000000	T38	072032	WRD.3 =	010536	WRT.DS	025214	
REGDMP	002404	T\$HILI=	000007	T39	072460	WRD.30=	011046	WRT.D1	024020	
REG.IN	015360	T\$LAST=	000000	T4	032336	WRD.31=	011054	WRT.D2	024332	
REG.1 =	012500	T\$LOLI=	000000	T40	073436	WRD.32=	011070	WRT.D3	024644	
REG.10=	012570	T\$LSYM=	010000	T41	074516	WRD.33=	011076	WRT.EE	023756	
REG.11=	012576	T\$NEST=	177777	T42	075324	WRD.34=	011102	WRT.EL	023714	
REG.12=	012604	T\$NSO =	000000	T43	076254	WRD.35=	011110	WRT.ER	022070	
REG.13=	012612	T\$NS1 =	000021	T44	076504	WRD.36=	011116	WRT.E1	023010	
REG.14=	012620	T\$PTHV=	000000	T45	076670	WRD.37=	011122	WRT.E2	023216	
REG.15=	012626	T\$PTNU=	000000	T46	077366	WRD.38=	011132	WRT.MR	022424	
REG.16=	012634	T\$SAVL=	177777	T47	077660	WRD.39=	011140	WRT.PA	022602	
REG.17=	012642	T\$SEGL=	177777	T48	100640	WRD.4 =	010544	WRT.PD	023450	
REG.18=	012652	T\$SUBN=	000000	T49	101376	WRD.40=	011150	WRT.RE	025324	
REG.19=	012660	T\$TAGL=	177777	T5	033006	WRD.41=	011154	WRT.TR	027426	
REG.2 =	012510	T\$TAGN=	010005	T50	102014	WRD.42=	011170	WT.DAT	015350	

W.C.SI	01574	\$T11	036302	\$T25	053472	\$T39	072046	\$T52	103216
W.ERR =	005.22	\$T12	036524	\$T26	054304	\$T4	031702	\$T53	104034
XOR.LN	021072	\$T13	037102	\$T27	055502	\$T40	072474	\$T54	104674
X\$ALWA=	000000	\$T14	040244	\$T28	056260	\$T41	073452	\$T55	105736
X\$FALS=	000040	\$T15	041476	\$T29	057244	\$T42	074532	\$T56	106626
X\$OFFS=	000400	\$T16	042730	\$T3	031474	\$T43	075340	\$T57	107756
X\$TRUE=	000020	\$T17	043572	\$T30	060356	\$T44	076270	\$T58	111250
\$END.L	113564 G	\$T18	045060	\$T31	061152	\$T45	076520	\$T59	111654
\$PATCH	003452 G	\$T19	045274	\$T32	062550	\$T46	076704	\$T6	033022
\$SAVE2	004540 G	\$T2	031150	\$T33	064306	\$T47	077402	\$T60	112322
\$SAVE3	004554 G	\$T20	046444	\$T34	065320	\$T48	077674	\$T61	112674
\$SAVE4	004572 G	\$T21	047444	\$T35	066302	\$T49	100654	\$T7	033512
\$SAVE5	004612 G	\$T22	051212	\$T36	066760	\$T5	032352	\$T8	034370
\$T1	030724	\$T23	051740	\$T37	070044	\$T50	101412	\$T9	035404
\$T10	035640	\$T24	052276	\$T38	070772	\$T51	102030		

. ABS. 113566 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 35594 WORDS (140 PAGES)

DYNAMIC MEMORY: 21558 WORDS (82 PAGES)

ELAPSED TIME: 00:16:01

ML11,ML11/-SP/CR:SYM=SVC/ML,CZMLAD.DOC,ML2AD,OTS,ML3AD,ML4AD

SYMBOL	VALUE	REFERENCES
ADR	= 000020 G	#94-2561
ARR.DA	= 013012	#98-2777 224-8947 230-9210 235-9514 243-9909 289-12284 290-12353 292-12438 336-14770
		342-15047 348-15402 349-15426 349-15450 350-15474 350-15494 361-16098 418-19076 424-19406
		447-20681 551-25985
ARR.IN	014004	#88-2240 *205-8023 *205-8043 307-13243 308-13292 308-13294 359-15984 369-16514 369-16523
		388-17530 389-17548 391-17695 392-17713 440-20280 440-20284
ARR.16	014016	#89-2251 *205-8008 *205-8024 *205-8044 308-13293
ASSEMB	= 000010	6-13 6-13
ASYN	= 012706	#98-2775 216-8527 220-8730 224-8929 229-9188 235-9494 236-9556 243-9900 249-10237
		250-10261 250-10281 253-10418 256-10574 256-10590 257-10610 261-10817 261-10832 262-10853
		262-10870 262-10886 263-10924 263-10943 264-10971 268-11190 269-11209 269-11226 269-11243
		269-11259 270-11297 270-11316 271-11344 275-11587 276-11606 276-11623 276-11640 277-11660
		277-11694 278-11717 278-11741 282-11937 283-11957 283-11975 283-11991 284-12012 284-12029
		295-12602 299-12830 300-12851 300-12869 301-12905 301-12938 302-12984 306-13191 306-13207
		307-13228 307-13261 308-13311 316-13715 316-13729 316-13742 316-13758 317-13778 318-13835
		318-13855 319-13912 320-13958 324-14134 324-14156 325-14184 329-14369 355-15775 374-16800
		383-17250 393-17797 441-20351 458-21211 459-21237 459-21275 460-21301 460-21316 463-21466
		528-24795 536-25244 547-25801 557-26344 558-26382 558-26394
A.CAL	015362	#89-2268 108-3283 *108-3284 *108-3285 111-3408 *483-22547 484-22582 *488-22798 489-22828
A.GEN	015370	#89-2271 111-3384 111-3408 *483-22544 484-22575 *488-22795 488-22817
BB.BB.	= 006334	#95-2592 558-26396
BB.INI	= 006242	#95-2591 558-26384
BB.VV.	= 006150	#95-2590 557-26346
BIT0	= 000001 G	#94-2545
BIT00	= 000001 G G	#94-2535
BIT01	= 000002 G G	#94-2534
BIT02	= 000004 G G	#94-2533
BIT03	= 000010 G G	#94-2532
BIT04	= 000020 G G	#94-2531
BIT05	= 000040 G G	#94-2530
BIT06	= 000100 G G	#94-2529
BIT07	= 000200 G G	#94-2528
BIT08	= 000400 G G	#94-2527
BIT09	= 001000 G G	#93-2522
BIT1	= 000002 G G	#94-2544
BIT10	= 002000 G G	#93-2521
BIT11	= 004000 G G	#93-2520
BIT12	= 010000 G G	#93-2519
BIT13	= 020000 G G	#93-2518
BIT14	= 040000 G G	#93-2517
BIT15	= 100000 G G	#93-2516
BIT2	= 000004 G G	#94-2543
BIT3	= 000010 G G	#94-2542
BIT4	= 000020 G G	#94-2541
BIT5	= 000040 G G	#94-2540
BIT6	= 000100 G G	#94-2539
BIT7	= 000200 G G	#94-2538
BIT8	= 000400 G G	#94-2537
BIT9	= 001000 G G	#94-2536
BLSDIV	004446 G	#35-1170 112-3480 514-24072 520-24397 562-26560
BLSGT1	003512 G	#11-169 424-19395 562-26559
BLSGT2	003634 G	#16-384 104-3098 108-3255 113-3506 299-12824 301-12899 302-12978 334-14654 336-14740

SYMBOL	CROSS REFERENCE VALUE		REFERENCES	CREF							
BL\$LAS	113556	G	348-15353	361-16070	368-16472	374-16773	380-17106	382-17223	389-17573	392-17737	399-18095
BL\$MOD	004460	G	406-18446	417-19045	418-19058	447-20663	478-22282	514-24096	521-24447	562-26558	
BL\$MUL	004222	G	#561-26522	561-26523	561-26527						
BL\$PU1	003776	G	#36-1201	113-3488	514-24111	520-24401	562-26560				
BL\$PU2	004072	G	#32-1022	526-24705	562-26560						
BL\$SHF	004472	G	#21-592	483-22532	562-26559						
			#27-829	111-3385	111-3392	111-3401	301-12932	418-19097	515-24128	521-24418	562-26559
			11-181	11-184	11-188	11-192	12-217	17-406	17-409	17-413	17-417
			17-438	21-595	21-599	21-605	22-632	27-838	27-842	27-848	28-880
			#37-1236								
BOE	= 000400	G	#94-2565								
B.CAL	015364		#89-2269	108-3289	*108-3290	*108-3291	111-3410	*483-22548	484-22583	*488-22799	489-22829
B.GEN	015372		#89-2272	111-3393	111-3410	*483-22545	484-22576	*488-22796	488-22818		
CAL.CR	017042		#107-3224	483-22550	488-22801						
CHIP.S	014012		#89-2248	*205-8022	*205-8042	551-25998					
C\$AU	= 000052		#6-13								
C\$AUTO	= 000061		#6-13								
C\$BRK	= 000022		#6-13								
C\$BSEG	= 000004		#6-13								
C\$BSUB	= 000002		#6-13								
C\$CEFG	= 000045		#6-13								
C\$CLCK	= 000062		#6-13								
C\$CLEA	= 000012		#6-13								
C\$CLOS	= 000035		#6-13								
C\$CLP1	= 000006		#6-13								
C\$CVEC	= 000036		#6-13								
C\$DCLN	= 000044		#6-13								
C\$DODU	= 000051		#6-13								
C\$DRPT	= 000024		#6-13								
C\$DU	= 000053		#6-13								
C\$EDIT	= 000003		#6-13								
C\$ERDF	= 000055		#6-13								
C\$ERHR	= 000056		#6-13								
C\$ERRO	= 000060		#6-13								
C\$ERSF	= 000054		#6-13								
C\$ERSO	= 000057		#6-13								
C\$ESCA	= 000010		#6-13								
C\$ESEG	= 000005		#6-13								
C\$ESUB	= 000003		#6-13								
C\$ETST	= 000001		#6-13								
C\$EXIT	= 000032		#6-13								
C\$GETB	= 000026		#6-13								
C\$GETW	= 000027		#6-13								
C\$GMAN	= 000043		#6-13								
C\$GPHR	= 000042		#6-13								
C\$GPLO	= 000030		#6-13								
C\$GPRI	= 000040		#6-13								
C\$INIT	= 000011		#6-13								
C\$INLP	= 000020		#6-13								
C\$MANI	= 000050		#6-13								
C\$MEM	= 000031		#6-13								
C\$MSG	= 000023		#6-13								

6-65

SYMBOL	VALUE	REFERENCES
CSOPEN	= 000034	#6-13
CSPNTB	= 000014	#6-13
CSPNTF	= 000017	#6-13
CSPNTS	= 000016	#6-13
CSPNTX	= 000015	#6-13
CSQIO	= 000377	#6-13
CSRDBU	= 000007	#6-13
CSREFG	= 000047	#6-13
CSRESE	= 000033	#6-13 #6-13
CSREVI	= 000003	#6-13 6-65
CSRFLA	= 000021	#6-13
CSRPT	= 000025	#6-13
CSSEFG	= 000046	#6-13
CSSPRI	= 000041	#6-13
CSSVEC	= 000037	#6-13
CSTPRI	= 000013	#6-13
DATA.L	= 013154	#98-2780
DAT.DM	016412	#101-2946 103-3054 348-15380 373-16711 373-16732 397-18014 398-18054 404-18369 405-18405 417-19006 446-20582 446-20619 514-24074 521-24425
DFPTBL	002360	G #6-119
DIAGMC	= 000000	6-13
DIVMOD	004264	#33-1091 35-1171 36-1202
DRIVE.	015354	#89-2262 *205-8019 *205-8039 236-9552 236-9565
DROP.C	015402	#89-2276 *203-7936 *204-7972 *204-7976 204-7979
DT.1	015764	#93-2485 526-24707 526-24709 527-24731 529-24840
DUMPER	026302	G #191-7313 196-7603 196-7622 199-7727 199-7744 209-8208 212-8370 216-8528 219-8672 220-8731 224-8930 224-8939 224-8948 229-9189 229-9198 230-9211 235-9495 235-9506 235-9515 236-9557 243-9901 243-9910 249-10238 250-10262 250-10282 253-10419 256-10575 256-10591 257-10611 261-10818 262-10837 262-10854 262-10871 262-10887 263-10925 263-10930 263-10944 264-10972 264-10977 268-11191 269-11210 269-11227 269-11244 269-11260 270-11298 270-11303 271-11321 271-11345 271-11350 271-11371 275-11588 276-11607 276-11624 276-11641 277-11661 277-11695 277-11700 278-11718 278-11742 278-11747 279-11772 282-11938 283-11958 283-11976 283-11992 284-12013 284-12030 289-12285 290-12354 292-12439 295-12603 300-12835 300-12852 300-12870 301-12906 301-12939 302-12985 306-13192 306-13208 307-13229 307-13262 308-13312 316-13716 316-13730 316-13743 316-13759 317-13779 318-13836 318-13856 319-13913 320-13959 324-14135 324-14157 325-14185 329-14370 329-14385 335-14698 336-14771 342-15048 348-15403 349-15427 349-15451 350-15475 350-15495 355-15776 361-16099 369-16529 374-16801 380-17130 383-17251 391-17675 393-17798 399-18128 400-18154 406-18465 411-18690 418-19077 424-19407 430-19713 430-19748 431-19790 435-20024 440-20290 441-20352 447-20682 453-20951 453-20976 454-20999 458-21212 459-21238 459-21276 460-21302 460-21317 463-21467 466-21590 469-21764 469-21799 472-21966 479-22306 484-22566 488-22808 494-23084 495-23114 496-23186 496-23215 502-23487 503-23518 508-23792 509-23823 515-24170 522-24481 528-24796 528-24811 529-24866 535-25185 536-25245 538-25354 542-25558 547-25797 547-25802 551-25986 557-26345 558-26383 558-26395
D1.TEM	= 013374	#98-2785 *104-3082 *355-15744 *360-16052 367-16403 *368-16457 *374-16759 *380-17092 *382-17209 *398-18077 *405-18428 *417-19028 *447-20649 *478-22274 *515-24129 515-24135 *520-24392 521-24422
D2.TEM	= 013376	#98-2786 *104-3083 *355-15745 *361-16057 367-16404 *368-16458 *374-16760 *380-17093 *382-17210 *398-18078 *406-18433 *417-19029 *447-20650 *478-22275 *515-24130 515-24136 *520-24393 521-24423
EF.CON	= 000036	G #94-2548
EF.NEW	= 000035	G #94-2549
EF.PWR	= 000034	G #94-2550
EF.RES	= 000037	G #94-2547

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
EF.STA	=	000040 G	#94-2546
EIG.FM	=	010364	#95-2626
ELV.FM	=	010466	#95-2629
ERRBLK		002160 G	#6-96
ERRMSG		002156 G	#6-96
ERRNBR		002154 G	#6-96
ERR1HR		002402 G	#6-161 93-2513 205-8011
ERRTYP		002152 G	#6-96
ERR.CH		017442	#110-3367 484-22555 488-22802
EVL	=	000004 G	#94-2559
ESEND	=	002100	#6-13
ESLOAD	=	000035	#6-13
E2.TEM	=	013400	#98-2787 6-65 *104-3084 *355-15746 *361-16058 367-16405 *368-16459 *374-16751 *380-17094 *382-17211 *398-18079 *406-18434 *417-19030 *447-20651 *478-22276 *515-24131 515-24137 *520-24372 *520-24373 521-24424
FIND.C		017660	#112-3475 494-23072 496-23176
FIRST.		020262	#120-3807 261-10811 268-11184 275-11581 323-14119 341-14276 354-15696 354-15718 381-17164
FIV.FM	=	010304	#95-2623 212-8376 236-9525 256-10581 257-10601 257-10617 261-10824 262-10843 262-10860 262-10877 263-10897 263-10936 264-10954 264-10983 268-11197 269-11216 269-11233 269-11250 270-11270 270-11309 271-11327 271-11356 275-11594 276-11613 276-11630 276-11647 277-11657 277-11706 278-11724 278-11753 283-11949 283-11964 283-11982 284-12003 284-12020 292-12445 300-12858 301-12912 302-12949 306-13198 306-13215 307-13235 307-13268 308-13318 316-13749 317-13769 317-13785 324-14141 324-14163 325-14191 355-15782 375-16811 406-18471 435-20030 440-20296 454-21005 460-21308 463-21473 497-23225 503-23524 509-23829
FMT.1	=	006442	#95-2593 220-8741 528-24821 538-25364 542-25572 547-25815
FMT.10	=	007116	#95-2602 335-14711
FMT.11	=	007170	#95-2603 213-8385
FMT.12	=	007224	#95-2604 324-14146 324-14168 325-14196
FMT.13	=	007254	#95-2605 337-14807
FMT.14	=	007330	#95-2606 393-17809
FMT.15	=	007370	#95-2607 431-19764
FMT.16	=	007436	#95-2608 225-8970 230-9229 236-9532 243-9925 289-12303 291-12376 292-12452 319-13922
FMT.17	=	007526	#95-2609 206-8073
FMT.18	=	007562	#95-2610
FMT.19	=	007632	#95-2611 484-22577 489-22823 529-24842
FMT.2	=	006506	#95-2594 219-8690 236-9566 253-10435 342-15067 411-18709 436-20045 537-25263
FMT.20	=	007702	#95-2612 484-22584 489-22830
FMT.21	=	007754	#95-2613 494-23094 495-23125 496-23196 497-23231 502-23498 503-23530 508-23803 509-23835
FMT.22	=	010026	#95-2614 516-24184 522-24491
FMT.23	=	010072	#95-2615 192-7383
FMT.24	=	010142	#95-2616 192-7390 192-7397 192-7404 193-7415 193-7422 193-7429 193-7436 193-7443 193-7450 193-7457 193-7464 194-7475 194-7483 194-7492 194-7499
FMT.25	=	010166	#95-2617 246-10078
FMT.26	=	010206	#95-2618 246-10088
FMT.3	=	006574	#95-2595
FMT.4	=	006620	#95-2596 295-12613
FMT.5	=	006650	#95-2597 406-18478 430-19728 431-19771 431-19805 479-22325
FMT.6	=	006752	#95-2598 448-20695
FMT.7	=	007002	#95-2599 302-12997 350-15511 458-21221 459-21247 460-21289
FMT.8	=	007030	#95-2600 551-26008 551-26019
FMT.9	=	007064	#95-2601 308-13277 308-13323 361-16108 441-20363 551-25994
FNC.1	=	012150	#97-2728 253-10421

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
ISCLN	=	000041	#6-13
ISDU	=	000041	#6-13
ISHRD	=	000041	#7-229 #7-250
ISINIT	=	000041	#6-13
ISMOD	=	000041	#6-13 6-39 #6-39 7-348 #7-348
ISMSG	=	000041	#6-13
ISPROT	=	000040	#6-13 #7-316
ISPTAB	=	000041	#6-13
ISPWR	=	000041	#6-13
ISRPT	=	000041	#6-13
ISSEFG	=	000041	#6-13
ISSETU	=	000041	#6-13
ISSET	=	000041	#7-277 #7-295
ISGRV	=	000041	#6-13
ISSUB	=	000041	#6-13
ISTST	=	000041	#6-13
JSJMP	=	000167	#6-13
LAST.B		020344	#122-3908 379-17042 379-17064 381-17182 459-21265
LAU		004704	#44-188 44-206
LAUTO		004660	#42-87 42-105
LCLEAN		113536	#560-26478 561-26502
LDU		004672	#43-136 43-154
LD.LNG		021354	#133-4443 367-16398 515-24134 521-24421
LIMIT		015400	#89-2275 *205-8013 *205-8015 212-8365 224-8919 229-9178 235-9480 242-9886 289-12278 290-12347 292-12432 307-13257 308-13307 324-14130 324-14152 324-14176 335-14693 342-15043 348-15398 349-15422 349-15446 350-15470 350-15490 406-18458 411-18685 418-19070 424-19402 430-19708 430-19743 431-19785 447-20675 453-20946 453-20971 458-21207 459-21233 463-21462 478-22295 484-22559 494-23077 494-23105 496-23181 496-23210 502-23480 502-23509 508-23785 509-23818 515-24165 522-24476 536-25238 538-25347 551-25981
LINIT		027650	#203-7927 206-8101
LOAD.S		016034	#98-2795 104-3111
LOE	=	040000	#94-2571
LOT	=	000010	#94-2560
LRPT		004646	#41-34 41-52
LST.AR		014020	#89-2252 *205-8009 *205-8029 *205-8030 *206-8055 *206-8056 307-13242 308-13291 359-15983 369-16513 369-16522 440-20279 440-20283 535-25200
LST.BL		014014	#89-2250 122-3910 *205-8010 *205-8035 *205-8036 *205-8037 *206-8063 *206-8064 *206-8065 317-13807 320-13952 381-17151 458-21185 459-21269 460-21288 463-21447 465-21577 535-25174 536-25223 537-25287 537-25304 538-25327 550-25958 557-26298 558-26376
LST.DU		015356	#89-2264 194-7487 *206-8069 *206-8071 212-8338
LSACP		002110	#6-65
LSAPT		002036	#6-65
LSAU		004706	6-65 #44-206
LSAUT		002070	#6-65
LSAUTO		004662	6-65 #42-105
LSCCP		002106	#6-65
LSCLEA		113546	6-65 #560-26497
LSCO		002032	#6-65
LSDEPO		002011	#6-65
LSDESC		002130	6-65 #6-87
LSDESP		002076	#6-65
LSDEVP		002060	#6-65

SYMBOL	VALUE		REFERENCES	CREF
LSDISP	002164	G	6-65	#6-103
LSDLY	002116	G	#6-65	103-3031 103-3058 103-3071 209-8196 212-8355 216-8516 220-8711 256-10563
			263-10904	270-11277 277-11674 282-11926 328-14354 334-14633 341-14982 341-15006 341-15019
			354-15701	354-15722 355-15737 360-16010 360-16033 360-16045 367-16410 367-16433 368-16450
			373-16716	373-16736 374-16752 379-17047 379-17068 380-17085 381-17186 382-17202 389-17553
			392-17718	397-18019 398-18058 398-18070 405-18378 405-18409 405-18421 417-19010 417-19021
			446-20587	446-20623 447-20642 478-22251 478-22267 514-24078 515-24141 521-24429 557-26316
			557-26333	
LSDTP	002040	G	#6-65	
LSDTYP	002034	G	#6-65	
LSDU	004674	G	6-65	#43-154
LSDUT	002072	G	#6-65	
LSDVTY	002122	G	6-65	#6-80
LSEF	002052	G	#6-65	
LSENV1	002044	G	#6-65	
LSERRT	002152	G	6-65	#6-96
LSETP	002102	G	#6-65	
LSEXP1	002046	G	#6-65	
LSEXP4	002064	G	#6-65	
LSEXP5	002066	G	#6-65	
LSHARD	002414	G	6-65	7-229 #7-229
LSHIME	002120	G	#6-65	
LSHPCP	002016	G	#6-65	
LSHPTP	002022	G	#6-65	
LSHW	002360	G	6-65	6-119 #6-119
LSICP	002104	G	#6-65	
LSINIT	030714	G	6-65	#206-8101
LSLADP	002026	G	#6-65	
LSLAST	= 113562	G	6-65	#561-26527
LSLOAD	002100	G	#6-65	
LSLUN	002074	G	#6-65	
LSMREV	002050	G	#6-65	
LSNAME	002000	G	#6-65	
LSPRIO	002042	G	#6-65	
LSPROT	003444	G	6-65	#7-316
LSPRT	002112	G	#6-65	
LSREPP	002062	G	#6-65	
LSREV	002010	G	#6-65	
LSRPT	004650	G	6-65	#41-52
LSSOFT	003122	G	6-65	7-277 #7-277
LSSPC	002056	G	#6-65	
LSSPCP	002020	G	#6-65	
LSSPTP	002024	G	#6-65	
LSSTA	002030	G	#6-65	
LSSW	002400	G	6-65	6-150 #6-150
LSTEST	002114	G	#6-65	
LSTIML	002014	G	#6-65	
LSUNIT	002012	G	#6-65	203-7920 204-7942 204-7987
L1000	002376		6-119	#6-139
L10001	002412		6-150	#6-165
L10002	002510		7-229	#7-250
L10003	003160		7-277	#7-295

ML11
 SYMBOL CROSS REFERENCE
 SYMBOL VALUE
 MB.DIS = 005570
 MEM.AR = 013052
 ML.DUT 016032

CREATED BY MACRO ON 30-MAR-82 AT 11:24

PAGE 9
 CREF

D 12

SEQ 0558

ML.LUN 016030

ML.REG 015404

REFERENCES	319-13914	103-3049	104-3120	169-6292	170-6317	173-6482	174-6507	178-6695	179-6723
#95-2585									
#98-2778									
#93-2509									
196-7586	199-7713	*205-8002	206-8078	209-8188	212-8348	216-8509	219-8661	219-8665	
219-8667	219-8683	219-8689	220-8702	220-8721	220-8726	220-8740	223-8898	229-9161	
234-9446	235-9468	240-9769	249-10227	250-10252	250-10272	253-10401	256-10556	261-10806	
263-10916	264-10963	268-11179	270-11289	271-11336	275-11576	277-11686	278-11733	282-11918	
289-12261	290-12330	291-12403	295-12587	299-12806	299-12813	299-12820	300-12884	301-12895	
301-12921	301-12928	302-12963	302-12967	302-12974	306-13181	307-13248	308-13298	315-13700	
317-13811	323-14114	328-14336	333-14606	340-14966	341-14997	347-15328	354-15688	354-15713	
359-15991	360-16022	366-16369	367-16422	373-16702	373-16728	379-17033	379-17059	381-17158	
381-17177	389-17539	392-17705	397-18006	398-18035	398-18047	404-18362	405-18390	405-18401	
410-18647	411-18663	416-18985	417-18998	422-19299	423-19564	423-19373	429-19668	434-19971	
435-20003	439-20247	440-20310	441-20329	445-20560	446-20599	446-20612	452-20897	452-20910	
458-21193	459-21260	463-21451	465-21571	468-21743	469-21785	472-21941	472-21952	477-22206	
483-22536	487-22762	493-23047	495-23149	501-23452	507-23753	513-24048	513-24057	515-24150	
520-24376	521-24457	526-24695	529-24857	535-25165	536-25208	537-25276	538-25320	538-25339	
541-25512	542-25533	546-25739	546-25757	550-25963	556-26278	557-26301			
#93-2508	196-7609	197-7632	199-7733	199-7750	*203-7935	*204-7941	204-7942	204-7945	
204-7981	*204-7985	*204-7986	204-7987	204-7990	206-8072	209-8223	213-8399	216-8541	
225-8989	230-9253	236-9573	243-9948	264-10995	272-11389	279-11786	292-12472	337-14819	
342-15082	356-15790	362-16124	369-16537	375-16815	400-18144	407-18501	412-18727	419-19120	
424-19437	432-19835	436-20060	441-20367	448-20721	479-22343	485-22612	489-22844		
#89-2278	101-2947	101-2948	101-2949	101-2950	103-3048	103-3051	103-3052	103-3055	
103-3067	103-3068	104-3082	104-3083	104-3084	104-3119	104-3122	104-3123	115-3620	
115-3631	115-3634	115-3635	118-3749	118-3757	118-3758	120-3808	120-3809	120-3810	
120-3811	121-3859	121-3860	121-3861	121-3862	122-3909	122-3910	122-3911	122-3912	
135-4569	135-4571	135-4572	135-4574	137-4668	137-4670	137-4671	137-4674	137-4675	
138-4728	138-4730	138-4731	138-4733	140-4828	140-4830	140-4831	140-4834	140-4835	
141-4888	141-4890	141-4891	141-4893	143-4987	143-4989	143-4990	143-4993	143-4994	
144-5047	144-5049	144-5050	144-5052	146-5147	146-5149	146-5150	146-5153	146-5154	
147-5200	147-5206	147-5208	147-5209	147-5211	147-5212	149-5302	149-5308	149-5310	
149-5311	149-5314	149-5315	149-5320	150-5362	150-5368	150-5370	150-5371	150-5373	
150-5374	152-5464	152-5470	152-5472	152-5473	152-5476	152-5477	152-5482	153-5537	
*153-5538	153-5544	153-5546	153-5547	154-5553	*154-5554	154-5555	156-5649	*156-5650	
156-5656	156-5658	156-5659	156-5662	156-5663	*156-5668	156-5669	157-5724	157-5725	
157-5731	157-5733	158-5738	158-5740	158-5741	158-5742	158-5743	160-5837	160-5838	
160-5844	160-5846	160-5847	160-5850	160-5851	160-5856	160-5857	163-5990	166-6125	
167-6170	167-6176	167-6178	167-6179	167-6181	167-6182	169-6285	169-6286	169-6287	
169-6288	169-6291	169-6294	169-6295	169-6296	169-6302	169-6304	170-6309	170-6312	
170-6316	170-6319	170-6320	171-6362	171-6368	171-6370	171-6371	171-6373	171-6374	
173-6475	173-6476	173-6477	173-6478	173-6481	173-6484	173-6485	173-6486	173-6492	
173-6494	173-6495	174-6502	174-6506	174-6509	174-6510	175-6566	*175-6567	175-6573	
175-6575	175-6576	176-6582	*176-6583	176-6584	*178-6687	178-6688	178-6689	178-6690	
178-6691	178-6694	178-6697	178-6698	178-6699	178-6705	179-6711	179-6712	179-6715	
179-6716	*179-6721	179-6722	179-6725	179-6726	182-6869	182-6871	182-6872	182-6875	
182-6876	192-7387	192-7388	192-7394	192-7395	192-7401	192-7402	192-7408	192-7409	
193-7419	193-7420	193-7426	193-7427	193-7433	193-7434	193-7440	193-7441	193-7447	
193-7448	193-7454	193-7455	193-7461	193-7462	194-7472	194-7473	194-7480	194-7481	
194-7489	194-7490	194-7496	194-7497	196-7585	196-7588	196-7589	196-7590	196-7591	
196-7592	196-7593	196-7594	196-7595	196-7596	196-7598	196-7613	196-7615	196-7617	
199-7712	199-7715	199-7716	199-7717	199-7718	199-7719	199-7720	199-7721	199-7722	

REFERENCES

199-7737	199-7739	*204-7963	206-8077	206-8080	206-8081	209-8187	209-8190	209-8191
209-8192	209-8193	209-8203	212-8347	212-8350	212-8351	212-8352	212-8362	213-8384
216-8508	216-8511	216-8512	216-8513	216-8523	219-8660	219-8663	219-8664	219-8665
219-8666	220-8706	220-8707	220-8708	220-8723	220-8724	220-8725	223-8897	223-8900
224-8905	225-8969	229-9160	229-9163	229-9164	230-9228	*234-9439	234-9445	234-9448
234-9449	234-9460	234-9461	234-9463	235-9470	235-9471	236-9531	*236-9551	236-9552
240-9768	240-9771	240-9772	243-9924	246-10050	246-10057	246-10060	246-10066	246-10076
249-10226	249-10229	249-10230	249-10231	249-10232	249-10233	250-10251	250-10254	250-10255
250-10256	250-10257	250-10271	250-10274	250-10275	250-10276	250-10277	252-10396	253-10403
253-10404	253-10407	253-10426	256-10555	256-10558	256-10559	256-10560	256-10570	256-10586
257-10606	261-10805	261-10808	261-10809	261-10812	261-10813	261-10828	262-10849	262-10866
262-10882	263-10911	263-10913	263-10915	263-10918	263-10919	263-10920	264-10960	264-10962
264-10965	264-10966	264-10967	268-11178	268-11181	268-11182	268-11185	268-11186	268-11201
269-11227	269-11239	269-11255	270-11284	270-11286	270-11288	270-11291	270-11292	270-11293
271-11333	271-11335	271-11338	271-11339	271-11340	271-11366	275-11575	275-11578	275-11579
275-11582	275-11583	276-11602	276-11619	276-11636	277-11656	277-11681	277-11683	277-11685
277-11688	277-11689	277-11690	278-11730	278-11732	278-11735	278-11736	278-11737	279-11767
282-11917	282-11920	282-11921	282-11922	282-11923	282-11933	283-11953	283-11971	283-11987
284-12008	284-12025	289-12260	289-12263	289-12264	289-12302	290-12329	290-12332	290-12333
291-12375	291-12402	291-12405	291-12406	291-12417	291-12418	292-12451	295-12586	295-12589
295-12590	295-12597	295-12598	299-12805	299-12808	299-12809	299-12810	299-12811	299-12812
300-12847	300-12864	300-12865	300-12883	301-12918	301-12919	301-12933	301-12934	302-12959
302-12960	302-12965	302-12966	306-13180	306-13183	306-13184	306-13185	306-13186	306-13187
306-13203	307-13224	307-13247	307-13250	307-13251	307-13252	307-13253	307-13254	308-13297
308-13300	308-13301	308-13302	308-13303	308-13304	315-13699	315-13702	315-13703	315-13704
315-13706	315-13707	316-13724	316-13725	316-13738	316-13754	317-13774	317-13810	317-13813
317-13814	318-13825	318-13829	318-13849	318-13851	319-13900	319-13901	319-13921	320-13948
323-14113	323-14116	323-14117	323-14120	323-14121	324-14127	324-14173	328-14335	328-14338
328-14339	328-14340	328-14347	328-14349	328-14351	328-14361	329-14380	333-14605	334-14612
334-14613	334-14615	334-14630	334-14640	340-14965	340-14968	340-14969	341-14975	341-14977
341-14978	341-14991	341-14992	341-14996	341-14999	341-15000	341-15001	341-15003	341-15013
341-15016	341-15026	347-15327	347-15330	347-15331	348-15381	348-15383	348-15384	348-15385
348-15386	354-15687	354-15690	354-15691	354-15692	354-15693	354-15694	354-15695	354-15697
354-15708	354-15712	354-15715	354-15716	354-15717	354-15719	355-15734	355-15744	355-15745
355-15746	359-15990	359-15993	359-15994	359-15995	359-15996	360-16001	360-16002	360-16003
360-16004	360-16005	360-16006	360-16017	360-16021	360-16024	360-16025	360-16026	360-16027
360-16028	360-16029	360-16030	360-16041	360-16042	360-16052	361-16057	361-16058	366-16368
366-16371	366-16372	366-16373	366-16374	366-16375	366-16376	367-16403	367-16404	367-16405
367-16406	367-16417	367-16421	367-16424	367-16425	367-16426	367-16427	367-16428	367-16429
367-16430	367-16442	367-16443	368-16457	368-16458	368-16459	373-16701	373-16704	373-16705
373-16708	373-16709	373-16710	373-16712	373-16723	373-16727	373-16730	373-16731	373-16733
373-16744	373-16745	374-16759	374-16760	374-16761	379-17032	379-17035	379-17036	379-17038
379-17039	379-17040	379-17041	379-17043	379-17054	379-17058	379-17061	379-17062	379-17063
379-17065	379-17077	380-17082	380-17092	380-17093	380-17094	381-17153	381-17154	381-17155
381-17157	381-17160	381-17161	381-17163	381-17165	381-17169	381-17176	381-17179	381-17180
381-17181	381-17183	382-17198	382-17199	382-17209	382-17210	382-17211	389-17538	389-17541
389-17542	389-17545	389-17546	389-17547	389-17549	389-17550	389-17560	390-17595	390-17603
390-17604	390-17609	390-17618	390-17619	390-17623	390-17631	390-17632	390-17637	391-17650
391-17651	391-17656	391-17657	391-17664	392-17704	392-17707	392-17708	392-17710	392-17711
392-17712	392-17714	392-17715	392-17725	392-17741	392-17754	393-17760	393-17762	393-17764
393-17766	393-17768	393-17774	393-17776	393-17782	393-17790	397-18005	397-18008	397-18009
397-18011	397-18012	397-18013	397-18015	397-18026	398-18034	398-18037	398-18038	398-18040

REFERENCES

398-18042	398-18043	398-18046	398-18049	398-18050	398-18055	398-18066	398-18067	398-18077
398-18078	398-18079	404-18361	404-18364	404-18365	404-18366	404-18367	404-18368	404-18370
405-18385	405-18389	405-18392	405-18393	405-18394	405-18397	405-18398	405-18400	405-18403
405-18404	405-18406	405-18417	405-18418	405-18428	406-18433	406-18434	410-18646	410-18649
410-18650	410-18651	410-18654	410-18655	411-18662	411-18665	411-18666	411-18675	411-18676
416-18984	416-18987	416-18988	416-18990	416-18991	416-18993	417-19000	417-19001	417-19007
417-19017	417-19018	417-19028	417-19029	417-19030	422-19298	422-19301	422-19302	423-19352
423-19353	423-19363	423-19366	423-19367	423-19369	423-19370	423-19372	423-19375	423-19376
429-19667	429-19670	429-19671	429-19684	429-19685	434-19970	434-19973	434-19974	435-19991
435-19992	435-20002	435-20005	435-20006	435-20008	435-20009	439-20246	439-20249	439-20250
439-20251	439-20252	439-20254	439-20255	439-20256	439-20257	439-20258	439-20259	439-20261
440-20267	440-20268	440-20269	440-20270	440-20271	440-20272	440-20274	440-20309	440-20312
440-20313	440-20314	440-20315	440-20317	440-20318	441-20323	441-20324	441-20325	441-20326
441-20328	441-20331	441-20332	441-20333	441-20335	441-20338	441-20339	441-20342	441-20343
441-20344	441-20345	441-20347	441-20361	445-20559	445-20562	445-20563	445-20575	445-20576
446-20581	446-20583	446-20594	446-20598	446-20601	446-20602	446-20606	446-20608	446-20609
446-20611	446-20614	446-20615	446-20620	447-20638	447-20639	447-20649	447-20650	447-20651
452-20896	452-20899	452-20900	452-20901	452-20904	452-20905	452-20909	452-20912	452-20913
452-20923	452-20924	452-20926	453-20935	453-20936	454-20994	458-21192	458-21195	458-21196
458-21197	458-21198	458-21199	458-21200	458-21201	458-21202	458-21204	459-21230	459-21259
459-21262	459-21263	459-21264	459-21266	459-21267	459-21271	460-21294	460-21296	460-21297
463-21450	463-21453	463-21454	463-21455	463-21456	463-21457	463-21458	463-21459	465-21570
465-21573	465-21574	465-21575	465-21576	465-21577	466-21582	466-21583	466-21585	468-21742
468-21745	469-21750	469-21751	469-21756	469-21757	469-21759	469-21775	469-21784	469-21787
469-21788	469-21789	469-21791	469-21792	469-21794	470-21818	470-21819	472-21940	472-21943
472-21944	472-21945	472-21948	472-21949	472-21951	472-21954	472-21955	472-21956	472-21958
472-21959	472-21961	477-22205	477-22208	477-22209	477-22210	477-22221	477-22222	477-22225
477-22226	477-22229	477-22230	477-22235	477-22238	477-22239	477-22241	477-22243	478-22263
478-22264	478-22274	478-22275	478-22276	483-22535	483-22538	483-22539	483-22541	483-22542
487-22761	488-22768	488-22769	488-22792	488-22793	493-23036	493-23039	493-23040	493-23046
493-23049	493-23050	494-23067	494-23068	494-23069	495-23148	495-23151	495-23152	496-23169
496-23170	496-23171	501-23419	501-23422	501-23423	501-23451	502-23458	502-23459	502-23471
502-23472	502-23473	502-23475	502-23477	507-23720	507-23723	507-23724	507-23752	507-23755
508-23760	508-23771	508-23773	508-23774	508-23776	508-23777	508-23778	508-23780	508-23782
513-24047	513-24050	513-24051	513-24053	513-24054	513-24056	513-24059	513-24060	514-24067
514-24068	514-24069	514-24075	514-24090	514-24105	515-24135	515-24136	515-24137	515-24138
515-24149	515-24152	515-24153	515-24155	515-24156	515-24157	520-24375	520-24378	520-24379
520-24388	520-24389	521-24422	521-24423	521-24424	521-24426	521-24441	521-24452	521-24456
521-24459	521-24460	521-24462	521-24463	521-24464	526-24694	526-24697	526-24698	526-24713
527-24718	527-24730	527-24733	527-24734	527-24736	527-24737	527-24738	527-24742	527-24749
527-24756	527-24763	528-24774	528-24785	528-24819	529-24856	529-24859	529-24860	529-24861
529-24876	535-25164	535-25167	535-25168	535-25174	535-25175	535-25176	535-25177	535-25178
535-25180	536-25207	536-25210	536-25211	536-25224	536-25227	536-25229	536-25230	536-25231
536-25232	536-25234	537-25275	537-25278	537-25279	537-25285	537-25286	537-25287	537-25288
537-25289	537-25293	537-25309	538-25319	538-25322	538-25323	538-25327	538-25331	538-25332
538-25333	538-25334	538-25338	538-25341	538-25342	538-25343	541-25511	541-25514	541-25515
541-25516	541-25517	541-25518	541-25519	541-25520	541-25523	541-25524	541-25528	542-25535
542-25536	542-25540	542-25543	542-25545	542-25546	542-25550	542-25568	546-25738	546-25741
546-25742	546-25748	546-25751	546-25752	546-25756	546-25759	546-25760	546-25768	546-25769
546-25771	546-25772	546-25775	546-25776	546-25778	546-25779	546-25783	547-25811	550-25962
550-25965	550-25966	550-25967	550-25968	550-25969	550-25970	550-25971	550-25972	550-25974
556-26277	556-26280	556-26281	557-26300	557-26303	557-26304	557-26305	557-26312	557-26329

SYMBOL	CROSS REFERENCE VALUE	REFERENCES	CREF
MSGH1	002510	557-26340 558-26368 558-26370 560-26478	
MSGH2	002556	7-242 #7-259	
MSGH3	002624	7-243 #7-260	
MSGH4	002670	7-244 #7-261	
MSGH5	002736	7-245 #7-262	
MSGH6	003004	7-246 #7-263	
MSGH7	003052	7-247 #7-264	
MSGS1	003160	7-248 #7-265	
MSGS2	003224	7-288 #7-304	
MSGS3	003270	7-289 #7-305	
MSGS4	003334	7-290 #7-306	
MSGS5	003400	7-291 #7-307	
MSDUMP	026312	7-292 #7-308	
NIB.SA	013374	191-7313 #192-7374	
		#88-2232 98-2785 98-2786 98-2787 99-2822 99-2828 99-2834 99-2840 99-2846	
		99-2852 100-2866 100-2872 100-2882 100-2891 125-4066 125-4073 125-4084 126-4096	
		126-4108 126-4115 126-4126 126-4134 126-4146 127-4158 130-4296 130-4299 130-4307	
		130-4309 130-4311 130-4318 130-4324 130-4326 130-4332 130-4336 *133-4460 *133-4467	
		*133-4471 *134-4483 *134-4484 *134-4487 *134-4494 *134-4498 *134-4506 *134-4507 *134-4511	
		*134-4519 *134-4520 479-22316	
NIN.FM	= 010410	#95-2627	
NO.INI	= 005470	#95-2584 320-13960	
ONEFIL	= 000001	#2-4 4-1152 5-1153 6-34 6-167 7-193 7-364	
ONEPAS	= 002406	#6-163 93-2512 204-7977	
ONE.FM	= 010240	#95-2619 192-7379 196-7605 196-7624 199-7729 199-7746 216-8530 315-13694 316-13718	
		317-13816 318-13838 318-13844 318-13858 319-13915 320-13961 556-26273 557-26307 557-26324	
		557-26347 558-26385 558-26397	
OP.NUM	014002	#88-2238 *205-8006 205-8025 205-8031 205-8045 206-8057 253-10405 253-10433 308-13289	
		389-17535 391-17696	
OSAPTS	= 000001	#6-13 #6-46 6-65	
OSAU	= 000001	#6-13 #6-46 6-65	
OSBGNR	= 000001	#6-13 #6-46 6-65	
OSBGNS	= 000001	#6-13 #6-46 6-65	
OSDU	= 000001	#6-13 #6-46 6-65	
OSERRT	= 000001	#6-13 #6-46 6-65	
OSGNSW	= 000001	#6-13 #6-46 6-65	
OSPOIN	= 000001	#6-13 #6-46 6-46 6-65	
OSSETU	= 000001	#6-13 #6-46 6-65	
PAR.DI	014010	#89-2247 *205-8000 249-10222 468-21730	
PD.TEM	015342	#89-2256 *103-3067 104-3092 *360-16041 361-16064 *367-16442 368-16466 *373-16744 374-16767	
		*379-17077 380-17100 *382-17198 382-17217 *389-17560 389-17567 *392-17725 392-17731 *398-18066	
		399-18089 *405-18417 406-18440 *417-19017 417-19040 *447-20638 447-20657 478-22262 *478-22263	
		514-24089 *514-24090 521-24440 *521-24441	
PHR.1	= 011610	#97-2713 261-10822 262-10841 264-10952 268-11195 269-11214 271-11325 275-11592 276-11611	
		278-11722 283-11980 295-12606 300-12838 300-12855 307-13233 308-13316 324-14161 329-14373	
		329-14388 349-15456 350-15480 350-15500 454-21003 460-21320 463-21471 466-21593 494-23085	
		496-23187	
PHR.10	= 012006	#97-2722 369-16531 391-17681 440-20301	
PHR.11	= 012020	#97-2723 528-24812 535-25186	
PHR.12	= 012036	#97-2724 516-24177	
PHR.13	= 012060	#97-2725 522-24484	
PHR.14	= 012104	#97-2726 192-7378	

G

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES	CREF
PHR.15	=	012136	#97-2727	246-10077 246-10087
PHR.2	=	011626	#97-2714	256-10579 262-10858 263-10934 264-10981 269-11231 270-11307 271-11354 276-11628
PHR.3	=	011644	277-11704	273-11751 300-12874 301-12910 302-12947 306-13196 316-13747 324-14139 460-21306
PHR.4	=	011676	#97-2715	216-8529
			#97-2716	205-8209 219-8678 224-8954 230-9217 236-9524 236-9558 289-12291 290-12360
			292-12444	347-15049 355-15780 375-16809 380-17133 399-18134 400-18160 430-19717 430-19752
			431-19794	435-20025 536-25246 551-25987
PHR.5	=	011714	#97-2717	212-8374 257-10599 257-10615 262-10855 262-10875 263-10895 269-11228 269-11248
			270-11268	272-11779 276-11645 277-11665 279-11776 283-11947 283-11959 283-11962 284-12001
			284-12018	306-13213 307-13266 316-13763 317-13783 325-14189 349-15412 349-15432 458-21214
			495-23115	547-25804
PHR.6	=	011726	#97-2718	262-10838 269-11211 276-11608 276-11625 283-11977 302-12990
PHR.7	=	011740	#97-2719	302-12986
PHR.8	=	011762	#97-2720	453-20955
PHR.9	=	011774	#97-2721	453-20980
PNT	=	001000	#94-2566	
PRI	=	002000	#94-2567	
PRI00	=	000000	#94-2558	
PRI01	=	000040	#94-2557	
PRI02	=	000100	#94-2556	
PRI03	=	000140	#94-2555	
PRI04	=	000200	#94-2554	
PRI05	=	000240	#94-2553	
PRI06	=	000300	#94-2552	
PRI07	=	000340	#94-2551	
PRSN		002400	#6-160	93-2512 245-10044
PTBL.P		014000	#88-2236	*204-7947 204-7948 204-7950 204-7951 204-7953 *204-7992 205-7997 205-7999
			205-8001	205-8003 205-8016 206-8066
PUP.BB	=	006060	#95-2589	557-26323
PWR.OF	=	005112	#94-2575	317-13815 557-26306
PWR.ON	=	005202	#94-2576	318-13843
P.AAA		004716	#66-960	94-2573
P.AAB		005012	#66-980	94-2574
P.AAC		005112	#66-1002	94-2575
P.AAD		005202	#67-1025	94-2576
P.AAE		005272	#67-1044	94-2577
P.AAF		005340	#67-1057	94-2578
P.AAG		005414	#68-1076	95-2583
P.AAH		005470	#68-1091	95-2584
P.AAI		005570	#68-1113	95-2585
P.AAJ		005652	#69-1134	95-2586
P.AAK		005722	#69-1148	95-2587
P.AAL		005764	#69-1160	95-2588
P.AAM		006060	#70-1184	95-2589
P.AAN		006150	#70-1203	95-2590
P.AAO		006242	#70-1223	95-2591
P.AAP		006334	#71-1247	95-2592
P.AAQ		006442	#71-1271	95-2593
P.AAR		006506	#71-1283	95-2594
P.AAS		006574	#72-1305	95-2595
P.AAT		006620	#72-1312	95-2596
P.AAU		006650	#72-1320	95-2597

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.AAV		006752	#72-1342 95-2598
P.AAW		007002	#73-1354 95-2599
P.AAX		007030	#73-1362 95-2600
P.AAY		007064	#73-1372 95-2601
P.AAZ		007116	#73-1381 95-2602
P.ABA		007170	#73-1395 95-2603
P.ABB		007224	#74-1409 95-2604
P.ABC		007254	#74-1417 95-2605
P.ABD		007330	#74-1432 95-2606
P.ABE		007370	#74-1443 95-2607
P.ABF		007436	#74-1456 95-2608
P.ABG		007526	#75-1479 95-2609
P.ABH		007562	#75-1489 95-2610
P.ABI		007632	#75-1503 95-2611
P.ABJ		007702	#76-1521 95-2612
P.ABK		007754	#76-1535 95-2613
P.ABL		010026	#76-1549 95-2614
P.ABM		010072	#76-1561 95-2615
P.ABN		010142	#77-1579 95-2616
P.ABO		010166	#77-1586 95-2617
P.ABP		010206	#77-1592 95-2618
P.ABQ		010240	#77-1601 95-2619
P.ABR		010246	#77-1603 95-2620
P.ABS		010256	#77-1606 95-2621
P.ABT		010270	#77-1610 95-2622
P.ABU		010304	#77-1614 95-2623
P.ABV		010322	#77-1619 95-2624
P.ABW		010342	#77-1625 95-2625
P.ABX		010364	#78-1635 95-2626
P.ABY		010410	#78-1642 95-2627
P.ABZ		010436	#78-1650 95-2628
P.ACA		010466	#78-1658 95-2629
P.ACB		010520	#78-1667 95-2630
P.ACC		010524	#78-1669 95-2631
P.ACD		010536	#78-1673 95-2632
P.ACE		010544	#78-1675 95-2633
P.ACF		010552	#78-1677 95-2634
P.ACG		010560	#78-1679 96-2639
P.ACH		010566	#78-1681 96-2640
P.ACI		010602	#79-1689 96-2641
P.ACJ		010616	#79-1693 96-2642
P.ACK		010630	#79-1697 96-2643
P.ACL		010640	#79-1700 96-2644
P.ACM		010650	#79-1703 96-2645
P.ACN		010660	#79-1706 96-2646
P.ACO		010664	#79-1708 96-2647
P.ACP		010676	#79-1712 96-2648
P.ACQ		010704	#79-1714 96-2649
P.ACR		010712	#79-1716 96-2650
P.ACS		010724	#79-1720 96-2651
P.ACT		010730	#79-1722 96-2652
P.ACU		010736	#79-1724 96-2653

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
P.AC		010744	#79-1726 96-2654
P.ACW		010752	#79-1728 96-2655
P.ACX		010766	#79-1732 96-2656
P.ACY		010774	#79-1734 96-2657
P.ACZ		011002	#79-1736 96-2658
P.ADA		011016	#80-1744 96-2659
F.ADB		011024	#80-1746 96-2660
F.ADC		011032	#80-1748 96-2661
P.ADD		011046	#80-1752 96-2662
P.ADE		011054	#80-1754 96-2663
P.ADF		011070	#80-1758 96-2664
P.ADG		011076	#80-1760 96-2665
P.ADH		011102	#80-1762 96-2666
P.ADI		011110	#80-1764 96-2667
P.ADJ		011116	#80-1766 96-2668
P.ADK		011122	#80-1768 96-2669
P.ADL		011132	#80-1771 96-2670
P.ADM		011140	#80-1773 96-2671
P.ADN		011150	#80-1776 96-2672
P.ADO		011154	#80-1778 96-2673
P.ADP		011170	#80-1782 96-2674
P.ADC		011176	#80-1784 96-2675
P.ADR		011206	#80-1787 96-2676
P.ADS		011214	#80-1789 96-2677
P.ADT		011226	#80-1793 96-2678
P.ADU		011240	#81-1801 96-2679
P.ADV		011250	#81-1804 96-2680
P.ADW		011260	#81-1807 96-2681
P.ADX		011270	#81-1810 96-2682
P.ADY		011276	#81-1812 96-2683
P.ADZ		011310	#81-1816 96-2684
P.AEA		011316	#81-1818 96-2685
P.AEB		011326	#81-1821 96-2686
P.AEC		011336	#81-1824 96-2687
P.AED		011344	#81-1826 96-2688
P.AEE		011352	#81-1828 96-2689
P.AEF		011362	#81-1831 96-2690
P.AEG		011374	#81-1835 97-2695
P.AEH		011400	#81-1837 97-2696
P.AEI		011416	#81-1842 97-2697
P.AEJ		011432	#81-1846 97-2698
P.AEK		011440	#81-1848 97-2699
P.AEL		011446	#81-1850 97-2700
P.AEM		011454	#82-1856 97-2701
P.AEN		011462	#82-1858 97-2702
P.AEO		011470	#82-1860 97-2703
P.AEP		011474	#82-1862 97-2704
P.AEQ		011502	#82-1864 97-2705
P.AER		011514	#82-1868 97-2706
P.AES		011524	#82-1871 97-2707
P.AET		011532	#82-1873 97-2708
P.AEU		011540	#82-1875 97-2709

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

CREATED BY MACRO ON 30-MAR-82 AT 11:24

PAGE 16
CREF

K 12

SEQ 0565

SYMBOL	VALUE	REFERENCES
P.AEV	011552	#82-1879 97-2710
P.AEW	011562	#82-1882 97-2711
P.AEX	011570	#82-1884 97-2712
P.AEY	011610	#82-1890 97-2713
P.AEZ	011626	#82-1895 97-2714
P.AFA	011644	#82-1900 97-2715
P.AFB	011676	#83-1913 97-2716
P.AFC	011714	#83-1918 97-2717
P.AFD	011726	#83-1922 97-2718
P.AFE	011740	#83-1926 97-2719
P.AFF	011762	#83-1932 97-2720
P.AFG	011774	#83-1936 97-2721
P.AFH	012006	#83-1940 97-2722
P.AFI	012020	#83-1944 97-2723
P.AFJ	012036	#83-1949 97-2724
P.AFK	012060	#83-1955 97-2725
P.AFL	012104	#83-1962 97-2726
P.AFM	012136	#84-1975 97-2727
P.AFN	012150	#84-1979 97-2728
P.AFO	012164	#84-1983 97-2729
P.AFP	012172	#84-1985 97-2730
P.AFQ	012200	#84-1987 97-2731
P.AFR	012216	#84-1992 97-2732
P.AFS	012226	#84-1995 97-2733
P.AFT	012234	#84-1997 97-2734
P.AFU	012244	#84-2000 97-2735
P.AFV	012260	#84-2004 97-2736
P.AFW	012272	#84-2008 97-2737
P.AFX	012302	#84-2011 97-2738
P.AFY	012322	#84-2017 97-2739
P.AFZ	012334	#85-2025 97-2740
P.AGA	012344	#85-2028 97-2741
P.AGB	012360	#85-2032 97-2742
P.AGC	012372	#85-2036 97-2743
P.AGD	012404	#85-2040 97-2744
P.AGE	012414	#85-2043 97-2745
P.AGF	012426	#85-2047 97-2746
P.AGG	012440	#85-2051 98-2751
P.AGH	012452	#85-2055 98-2752
P.AGI	012464	#85-2059 98-2753
P.AGJ	012500	#85-2063 98-2754
P.AGK	012510	#85-2066 98-2755
P.AGL	012516	#85-2068 98-2756
P.AGM	012524	#85-2070 98-2757
P.AGN	012532	#85-2072 98-2758
P.AGO	012540	#85-2074 98-2759
P.AGP	012546	#86-2080 98-2760
P.AGQ	012554	#86-2082 98-2761
P.AGR	012562	#86-2084 98-2762
P.AGS	012570	#86-2086 98-2763
P.AGT	012576	#86-2088 98-2764
P.AGU	012604	#86-2090 98-2765

SYMBOL	CROSS REFERENCE VALUE	REFERENCES
P.AGV	012612	#86-2092 98-2766
P.AGW	012620	#86-2094 98-2767
P.AGX	012626	#86-2096 98-2768
P.AGY	012634	#86-2098 98-2769
P.AGZ	012642	#86-2100 98-2770
P.AHA	012652	#86-2103 98-2771
P.AHB	012660	#86-2105 98-2772
P.AHC	012666	#86-2107 98-2773
P.AHD	012676	#86-2110 98-2774
P.AHE	012706	#86-2113 98-2775
P.AHF	012750	#86-2125 98-2776
P.AHG	013012	#87-2141 98-2777
P.AHH	013052	#87-2152 98-2778
P.AHI	013114	#87-2164 98-2779
P.AHJ	013154	#87-2175 98-2780
P.AHK	013214	#87-2186 98-2781
P.AHL	013250	#88-2200 98-2782
P.AHM	013302	#88-2209 98-2783
P.AHN	013330	#88-2217 98-2784
P.CAL	015366	#89-2270 108-3277 *108-3278 *108-3279 111-3406 *483-22549 484-22581 *488-22800 489-22827
P.CNT	015376	#89-2274 *212-8336 *212-8364 212-8365 *223-8891 *224-8918 224-8919 *229-9153 *229-9177 229-9178 *234-9438 *235-9479 235-9480 *240-9766 *242-9885 242-9886 *288-12252 *289-12277 289-12278 *290-12346 290-12347 *292-12431 292-12432 *307-13241 *307-13256 307-13257 *308-13288 *308-13306 308-13307 *323-14105 *324-14129 324-14130 *324-14151 324-14152 *324-14175 324-14176 *333-14604 *335-14692 335-14693 *340-14964 *342-15042 342-15043 *347-15323 *348-15397 348-15398 *349-15421 349-15422 *349-15445 349-15446 *350-15469 350-15470 *350-15489 350-15490 *404-18355 *406-18457 406-18458 *410-18641 *411-18684 411-18685 *415-18934 *418-19069 418-19070 *422-19297 *424-19401 424-19402 *429-19665 *430-19707 430-19708 *430-19742 430-19743 *431-19784 431-19785 *445-20557 *447-20674 447-20675 *452-20895 *453-20945 453-20946 *453-20970 453-20971 *458-21184 *458-21206 458-21207 *459-21232 459-21233 *463-21446 *463-21461 463-21462 *477-22200 *478-22294 478-22295 *483-22507 *484-22558 484-22559 *493-23035 *494-23076 494-23077 *494-23104 494-23105 *496-23180 496-23181 *496-23209 496-23210 *501-23418 *502-23479 502-23480 *502-23508 502-23509 *507-23719 *508-23784 508-23785 *509-23817 509-23818 *513-24037 *515-24164 515-24165 *520-24371 *522-24475 522-24476 *535-25162 *536-25237 536-25238 *538-25346 538-25347 *550-25957 *550-25976 551-25981
P.GEN	015374	#89-2273 111-3378 111-3406 *483-22546 484-22574 *488-22797 488-22816
RAS.IN	015346	#89-2259 *205-8021 *205-8041 369-16509 439-20243 439-20253
RD.CS1	021762	#137-4661 189-7216 242-9847
RD.DA	022316	#143-4980 189-7226 242-9861
RD.DAT	015352	#89-2261 *137-4676 137-4677 *140-4836 140-4837 *143-4995 143-4996 *146-5155 146-5156 *149-5316 149-5317 *152-5478 152-5479 *156-5664 156-5665 *160-5852 160-5853 *163-5990 163-5991 *166-6125 166-6126 *170-6312 170-6313 *174-6502 174-6503 *179-6717 179-6718 *182-6877 182-6878 225-8963 230-9222 236-9529 243-9918 289-12296 290-12365 292-12449 319-13919
RD.DS	025216	#182-6862 190-7270
RD.D1	024104	#169-6283 190-7275
RD.D2	024416	#173-6473 190-7280
RD.D3	024744	#178-6685 190-7285
RD.EE	023760	#166-6123 189-7256
RD.EL	023716	#163-5988 189-7261
RD.ER	022140	#140-4821 189-7221 242-9854
RD.E1	023074	#152-5462 189-7236 242-9875

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

CREATED BY MACRO ON 30-MAR-82 AT 11:24

PAGE 19
CREF

N 12

SEQ 0568

REFERENCES

SVCINS = 177777

SVCSUB = 177777
SVCTAG = 177777

SVCTST = 177777
SYNC = 012750

SSLSYM = 010000
TEN.FM = 010436
THR.FM = 010256

TIME.O = 013330
TRBLE. = 013250
TST.LN 020376

TWO.FM = 010246
TSARGC = 000002

TSCODE = 004130

6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-80	6-80	6-87	6-87	6-87	6-96	6-96	6-96	6-96	6-80
6-103	6-103	6-119	6-119	6-119	6-119	6-119	6-119	6-119	6-103
6-150	6-150	6-150	6-150	6-150	7-229	7-229	7-229	7-229	6-150
7-277	7-277	7-316	7-316	7-316					7-277
#6-13	#6-19	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65	6-65
6-103	6-103	6-103	6-103	6-103	6-80	6-80	6-87	6-87	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103	6-103
7-229	7-242	7-242	7-242	7-242	7-242	7-242	7-242	7-242	6-119
7-243	7-244	7-244	7-244	7-244	7-243	7-243	7-243	7-243	6-150
7-245	7-245	7-246	7-246	7-246	7-244	7-245	7-245	7-245	7-243
7-247	7-248	7-248	7-248	7-248	7-247	7-247	7-247	7-247	7-245
7-289	7-289	7-289	7-289	7-290	7-277	7-288	7-288	7-288	7-247
7-292	7-292	7-292	7-292	7-295	7-290	7-291	7-291	7-291	7-288
#6-13	#6-21								7-291
#6-13	#6-23	6-139	6-139	6-139	6-165	6-165	6-165	6-165	7-250
7-250	7-250	7-295	7-295	7-295					
#6-13	#6-20								
#98-2776	224-8938	229-9197	235-9505	263-10929	264-10976	270-11302	271-11349	277-11699	
278-11746	329-14384	399-18127	406-18464	411-18689	430-19712	430-19747	431-19789	435-20023	
453-20950	453-20975	454-20998	466-21589	469-21763	469-21798	479-22305	484-22565	488-22807	
494-23083	495-23113	496-23185	496-23214	502-23486	503-23517	508-23791	509-23822	515-24169	
522-24480	528-24810	529-24865	535-25184	538-25353	542-25557	547-25796			
#6-13	#6-139	#6-165	#7-250	#7-295					
#95-2628									
#95-2621	209-8212	220-8735	243-9914	250-10246	250-10266	250-10286	361-16103	369-16533	
381-17138	383-17255	440-20303	447-20686	458-21216	459-21242	460-21284	468-21735	484-22570	
488-22812	494-23088	496-23190	516-24178	522-24485	528-24815	535-25189	538-25358	547-25806	
#98-2784									
#98-2782	472-21965								
#125-4047	355-15752	361-16078	368-16480	374-16781	380-17114	382-17231	418-19066	447-20671	
478-22291									
#95-2620	236-9560	251-10302	253-10422	316-13733	381-17144	536-25248	551-25989		
#6-65	6-65	#6-65	6-65	6-65	#6-65	6-65	6-65	#6-65	
6-65	6-65	#6-65	6-65	6-65	#6-65	6-65	6-65	6-65	
#7-242	7-242	#7-242	7-242	#7-242	7-242	#7-243	7-243	#7-243	

SYMBOL	VALUE	REFERENCES
TSEHRN	= 000000	#7-243 7-243 #7-244 7-244 #7-244 7-244 #7-244 7-244 #7-244 7-244
TSEXCP	= 000000	#7-245 7-245 #7-245 7-245 #7-245 7-245 #7-245 7-245 #7-245 7-245
T\$FREE	113562 G	7-246 #7-246 7-246 #7-247 7-247 #7-247 7-247 #7-247 7-247
T\$GMAN	= 000000	#7-248 7-248 #7-248 7-248 #7-248 7-248 #7-248 7-248 #7-248 7-248
T\$HILI	= 000007	7-288 #7-288 7-288 #7-289 7-289 #7-289 7-289 #7-289 7-289
T\$LAST	= 000000	#7-290 7-290 #7-290 7-290 #7-290 7-290 #7-290 7-290 #7-290 7-290
T\$LOLI	= 000000	7-291 #7-291 7-291 #7-292 7-292 #7-292 7-292 #7-292 7-292
T\$LSYM	= 010000	#6-13 7-242 #7-243 7-243 #7-244 7-244 #7-245 7-245 #7-247 7-247
T\$NEST	= 177777	7-247 561-26522 561-26523 #561-26524
T\$NSO	= 000000	#6-13 7-242 #7-243 7-243 #7-244 7-244 #7-245 7-245 #7-247 7-247
T\$NST	= 000021	7-247 #6-13 6-139 #6-39 6-139 #6-150 6-150 #6-150 6-150 #6-150 6-150
T\$PTHV	= 000000 G	6-139 #6-139 6-139 #6-150 6-150 #6-150 6-150 #6-150 6-150
T\$PTNU	= 000000	7-229 #7-229 7-229 #7-250 7-250 #7-250 7-250 #7-250 7-250
T\$SAVL	= 177777	7-277 7-295 #7-322 7-322 #7-322 7-322 #7-322 7-322
T\$SEGL	= 177777	7-322 #6-39 7-348 #6-150 6-165 #7-229 7-250 #7-277 7-295 #7-316 7-316 #7-348 7-348
T\$SUBN	= 000000	#6-39 6-139 #6-150 6-165 #7-229 7-250 #7-277 7-295 #7-316 7-316 #7-348 7-348
T\$TAGL	= 177777	#6-119 6-119 #6-119 6-150 6-150 #6-150 7-229 7-229
T\$TAGN	= 010005	#7-229 7-277 #7-277 7-316 7-316 #7-316 7-229 7-229
T\$TEMP	= 000000	#6-103 6-103 #6-103 6-103 #6-103 6-103 #6-103 6-103 #6-103 6-103

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

CREATED BY MACRO ON 30-MAR-82 AT 11:24

PAGE 21
CREF

C 13

SEQ 0570

REFERENCES

#6-103	6-103	6-103	#6-103	6-103	6-103	#6-103	6-103	6-103
#6-103	6-103	6-103	#6-103	6-103	6-103	#6-103	6-103	6-103
#6-103	6-103	6-103	#6-103	#6-139	6-139	#6-165	6-165	#7-242
7-242	#7-242	7-242	#7-242	7-242	#7-243	7-243	#7-243	7-243
#7-243	7-243	#7-244	7-244	#7-244	7-244	#7-244	7-244	#7-245
7-245	#7-245	7-245	#7-245	7-245	#7-246	7-246	#7-246	7-246
#7-246	7-246	#7-247	7-247	#7-247	7-247	#7-247	7-247	#7-248
7-248	#7-248	7-248	#7-248	7-248	#7-250	7-250	#7-288	7-288
#7-288	7-288	#7-288	7-288	#7-289	7-289	#7-289	7-289	#7-289
7-289	#7-290	7-290	#7-290	7-290	#7-290	7-290	#7-291	7-291
#7-291	7-291	#7-291	7-291	#7-292	7-292	#7-292	7-292	#7-292
7-292	#7-295	7-295	#7-322	7-322	#7-348	7-348		

TSTEST	=	000000	
TSTSTM	=	177777	
TSTSTS	=	000000	
TSSHAR	=	010002	
TSSHW	=	010000	
TSSPRO	=	010004	
TSSSOF	=	010003	
TSSSW	=	010001	
T.21	=	004716	
T.61	=	005012	
T1		031134	G
T10		036266	G
T11		036510	G
T12		037066	G
T13		040230	G
T14		041462	G
T15		042714	G
T16		043556	G
T17		045044	G
T18		045260	G
T19		046430	G
T2		031460	G
T20		047430	G
T21		051176	G
T22		051724	G
T23		052262	G
T24		053456	G
T25		054270	G
T26		055466	G
T27		056244	G
T28		057230	G
T29		060342	G
T3		031666	G
T30		061136	G
T31		062534	G
T32		064272	G
T33		065304	G
T34		066266	G
T35		066744	G
T36		070030	G

#6-13			
#6-13			
#6-13			
#7-229	7-229	7-250	
#6-119	6-119	6-139	
#7-316			
#7-277	7-277	7-295	
#6-150	6-150	6-165	
#94-2573	315-13693		
#94-2574	556-26272		
6-103	#210-8249		
6-103	#251-10325		
6-103	#254-10464		
6-103	#257-10645		
6-103	#265-11021		
6-103	#272-11411		
6-103	#279-11808		
6-103	#285-12070		
6-103	#293-12498		
6-103	#296-12647		
6-103	#303-13033		
6-103	#213-8421		
6-103	#309-13356		
6-103	#321-13995		
6-103	#325-14231		
6-103	#330-14424		
6-103	#338-14843		
6-103	#343-15108		
6-103	#351-15546		
6-103	#356-15813		
6-103	#362-16150		
6-103	#370-16564		
6-103	#217-8565		
6-103	#375-16838		
6-103	#383-17289		
6-103	#394-17844		
6-103	#400-18185		
6-103	#407-18528		
6-103	#412-18755		
6-103	#419-19142		

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
T37		070756 G	6-103 #425-19463
T38		072032 G	6-103 #432-19857
T39		072460 G	6-103 #436-20081
T4		032336 G	6-103 #221-8773
T40		073436 G	6-103 #442-20393
T41		074516 G	6-103 #448-20743
T42		075324 G	6-103 #454-21035
T43		076251 G	6-103 #461-21352
T44		076504 G	6-103 #464-21506
T45		076670 G	6-103 #466-21618
T46		077366 G	6-103 #470-21843
T47		077660 G	6-103 #473-22004
T48		100640 G	6-103 #480-22367
T49		101376 G	6-103 #485-22634
T5		033006 G	6-103 #225-9011
T50		102014 G	6-103 #489-22865
T51		103202 G	6-103 #497-23266
T52		104020 G	6-103 #504-23572
T53		104660 G	6-103 #510-23877
T54		105722 G	6-103 #516-24223
T55		106612 G	6-103 #523-24530
T56		107742 G	6-103 #530-24905
T57		111234 G	6-103 #539-25401
T58		111640 G	6-103 #543-25605
T59		112306 G	6-103 #548-25849
T6		033476 G	6-103 #231-9279
T60		112660 G	6-103 #552-26052
T61		113522 G	6-103 #559-26426
T7		034354 G	6-103 #237-9599
T8		035370 G	6-103 #244-9974
T9		035624 G	6-103 #247-10114
UAM	=	000200 G	#94-2564
UNS.ER	=	005272	#94-2577 318-13837
VV.CLE	=	005414	#95-2583 316-13717
VV.NOT	=	005340	#94-2578 316-13732
VV.SET	=	005764	#95-2588 318-13857
WC.ERR	=	005652	#95-2586 196-7604 196-7623
WRD.1	=	010520	#95-2630 256-10580 261-10823 262-10839 262-10856 263-10935 264-10982 268-11196 269-11212
			269-11229 270-11308 271-11355 275-11593 276-11609 276-11626 277-11705 278-11752 283-11948
			283-11960 283-11978 306-13197 316-13748 324-14140
WRD.10	=	010630	#96-2643 243-9911 335-14703 361-16100 383-17252 393-17799 406-18468 411-18695 418-19081
			424-19411 447-20683 453-20952 453-20977 463-21468 473-21975 479-22307 502-23490 508-23795
			528-24799 538-25355 542-25561 547-25803
WRD.11	=	010640	#96-2644 256-10578 257-10598 257-10614 263-10933 264-10951 264-10980 270-11306 271-11324
			271-11353 277-11703 278-11721 278-11750 283-11946 300-12837 300-12854 300-12873 301-12909
			301-12942 302-12989 306-13195 306-13212 316-13746 316-13762 317-13782 324-14138 324-14160
			329-14372 329-14387 454-21002 460-21305 460-21319 463-21470 466-21592
WRD.12	=	010650	#96-2645 212-8373 219-8677 224-8953 230-9216 235-9519 261-10821 262-10874 263-10894
			268-11194 269-11247 270-11267 272-11378 275-11591 276-11644 277-11664 279-11775 284-12000
			284-12017 289-12290 290-12359 292-12443 295-12605 307-13232 307-13265 308-13315 325-14188
			335-14702 355-15779 375-16808 391-17679 399-18132 400-18158 406-18467 411-18694 418-19080
			424-19410 430-19716 430-19751 431-19793 473-21974 496-23217 503-23520 509-23825

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
WRD.13	=	010660	#96-2646 219-8674 284-12032 380-17132
WRD.14	=	010664	#96-2647 220-8732 253-10420 284-12033 369-16532 391-17682 440-20302 441-20353 459-21239
			459-21277 528-24797 529-24867 542-25559
WRD.15	=	010676	#96-2648 300-12839 300-12875 301-12911 302-12948 302-2991
WRD.16	=	010704	#96-2649 300-12856 307-13234
WRD.17	=	010712	#96-2650 301-12941 302-12988 460-21304
WRD.18	=	010724	#96-2651
WRD.19	=	010730	#96-2652 256-10576 257-10596 257-10612 261-10819 262-10872 262-10888 263-10931 264-10949
			264-10978 268-11192 269-11245 270-11265 270-11304 271-11322 271-11351 271-11372 275-11589
			276-11642 277-11662 277-11701 278-11719 278-11748 279-11773 282-11939 283-11993 284-12014
			284-12034 300-12871 301-12907 306-13193 306-13209 307-13230 307-13263 308-13313 316-13744
			316-13760 317-13780 324-14136 324-14158 325-14186 342-15050 399-18129 400-18155 411-18691
			418-19078 424-19408 430-19714 430-19749 431-19791 435-20027 453-20953 453-20978 454-21001
WRD.2	=	010524	#95-2631 262-10842 262-10859 264-10953 269-11215 269-11232 271-11326 276-11612 276-11629
			278-11723 283-11963 283-11981
WRD.20	=	010736	#96-2653 272-11380 279-11777 329-14389
WRD.21	=	010744	#96-2654 329-14371 329-14374 329-14386
WRD.22	=	010752	#96-2655 355-15781 375-16810 399-18135
WRD.23	=	010766	#96-2656 399-18131 400-18157 411-18693 430-19719 430-19754 431-19796 435-20029 469-21768
			470-21807
WRD.24	=	010774	#96-2657 349-15408 349-15428 349-15452 350-15476 350-15496 406-18470 411-18697 479-22309
WRD.25	=	011002	#96-2658 406-18469 411-18696
WRD.26	=	011016	#96-2659 466-21594
WRD.27	=	011024	#96-2660 458-21215 460-21307 460-21318 460-21321
WRD.29	=	011032	#96-2661 458-21213
WRD.3	=	010536	#95-2632 257-10600 262-10876 269-11249 276-11646 284-12002 306-13214 317-13768 324-14162
WRD.30	=	011046	#96-2662 463-21469 463-21472
WRD.31	=	011054	#96-2663 459-21240 459-21278
WRD.32	=	011070	#96-2664 349-15411 349-15431 349-15455 350-15479 350-15499
WRD.33	=	011076	#96-2665 349-15409 349-15429 349-15453 350-15477 350-15497
WRD.34	=	011102	#96-2666 349-15413 349-15433 349-15457 350-15481 350-15501
WRD.35	=	011110	#96-2667 335-14700 342-15052 361-16102 551-25988
WRD.36	=	011116	#96-2668 342-15051
WRD.37	=	011122	#96-2669 219-8675 220-8733 243-9912 361-16101 391-17677 393-17800 468-21732
WRD.38	=	011132	#96-2670 243-9913
WRD.39	=	011140	#96-2671 393-17802 430-19718 430-19753 431-19795 435-20026
WRD.4	=	010544	#95-2633 257-10616 263-10896 270-11269 277-11666 284-12019 307-13267 308-13317 317-13784
			325-14190
WRD.40	=	011150	#96-2672 336-14774 337-14790
WRD.41	=	011154	#96-2673 447-20685
WRD.42	=	011170	#96-2674 336-14775 337-14791
WRD.43	=	011176	#96-2675 262-10840 262-10857 269-11213 269-11230 276-11610 276-11627 283-11961 283-11979
WRD.44	=	011206	#96-2676
WRD.45	=	011214	#96-2677 335-14701 355-15778
WRD.46	=	011226	#96-2678 337-14781 337-14793 447-20684
WRD.47	=	011240	#96-2679 337-14792
WRD.48	=	011250	#96-2680 374-16802
WRD.49	=	011260	#96-2681
WRD.5	=	010552	#95-2634 249-10241 349-15454 350-15478 350-15498 484-22569 488-22811
WRD.50	=	011270	#96-2682 383-17254 418-19082 424-19412 440-20293 441-20355
WRD.51	=	011276	#96-2683 337-14780 383-17253
WRD.52	=	011310	#96-2684 212-8372 224-8950 230-9213 235-9518 289-12287 290-12356 292-12442 528-24813

ML11
SYMBOL CROSS REFERENCE
SYMBOL VALUE

MACRO ON 30-MAR-82 AT 11:24

PAGE 24
CREF

F 13

SEQ 0573

SYMBOL	VALUE	REFERENCES
WRD.53	= 011316	#96-2685 535-25187
WRD.54	= 011326	#96-2686 250-10296
WRD.55	= 011336	#96-2687 337-14782
WRD.56	= 011344	#96-2688 337-14783
		440-20291 224-8949 230-9212 235-9516 289-12286 290-12355 292-12440 381-17142 391-17676
WRD.57	= 011352	#96-2689
WRD.58	= 011362	#96-2690
WRD.59	= 011374	#97-2695 454-21004
WRD.6	= 010560	#96-2639 250-10265 250-10285 349-15410 349-15430 469-21767 470-21806
WRD.60	= 011400	#97-2696 440-20292 441-20354
WRD.61	= 011416	#97-2697 272-11381 279-11778 495-23118 496-23220 503-23523 509-23828
WRD.62	= 011432	#97-2698 212-8375
WRD.63	= 011440	#97-2699
WRD.64	= 011446	#97-2700 479-22310 484-22568 488-22810
WRD.65	= 011454	#97-2701
WRD.67	= 011462	#97-2702 496-23216 502-23491 503-23519 509-23824 528-24800 542-25562
WRD.68	= 011470	#97-2703
WRD.69	= 011474	#97-2704 528-24798 542-25560
WRD.7	= 010566	#96-2640 249-10240 250-10264 250-10284 250-10297 468-21733 469-21766 469-21801
WRD.70	= 011502	#97-2705
WRD.71	= 011514	#97-2706 538-25356
WRD.72	= 011524	#97-2707 547-25805
WRD.73	= 011532	#97-2708 479-22308
WRD.74	= 011540	#97-2709 494-23087 495-23117 496-23189 496-23219 503-23522 509-23827
WRD.75	= 011552	#97-2710 494-23086 495-23116 496-23188 496-23218 503-23521 509-23826
WRD.76	= 011562	#97-2711 502-23489 508-23794 515-24172 522-24483
WRD.77	= 011570	#97-2712 508-23796
WRD.8	= 010602	#96-2641 250-10283 469-21800 484-22567 488-22809
WRD.9	= 010616	#96-2642 249-10239 250-10263 399-18133 400-18159 469-21765 502-23488 508-23793 515-24171
		522-24482
WRT.CH	027156	#196-7584 320-13947 558-26367
WRT.CS	021712	#135-4563 185-7017 240-9777 241-9809
WRT.DA	022246	#141-4882 185-7027 241-9787 241-9817
WRT.DS	025214	#180-6775 186-7071
WRT.D1	024020	#167-6169 186-7076
WRT.D2	024332	#171-6361 186-7081
WRT.D3	024644	#175-6565 186-7086
WRT.EE	023756	#164-6041 185-7057
WRT.EL	023714	#161-5904 186-7066
WRT.ER	022070	#138-4722 185-7022 240-9780 241-9813
WRT.E1	023010	#150-5361 185-7037 241-9793 241-9825
WRT.E2	023216	#153-5536 185-7042 241-9796
WRT.MR	022424	#144-5041 185-7032
WRT.PA	022602	#147-5199 185-7047 241-9790 241-9821
WRT.PD	023450	#157-5723 185-7052 446-20605
WRT.RE	025324	#184-6990 224-8910 229-9169 234-9454 289-12269 290-12338 291-12411
WRT.TR	027426	#199-7711 317-13805 557-26296
WT.DAT	015350	#89-2260 *137-4673 137-4677 *140-4833 140-4837 *143-4992 143-4996 *146-5152 146-5156
		*149-5313 149-5317 *152-5475 152-5479 *156-5661 156-5665 *160-5849 160-5853 *163-5989
		163-5991 *166-6124 166-6126 *170-6311 170-6313 *174-6501 174-6503 *179-6714 179-6718
		*182-6874 182-6878 225-8964 230-9223 243-9919 289-12297 291-12370

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
W.C.SI		015344	#89-2257 *205-8020 *205-8040 439-20254 440-20267 441-20340
W.ERR	=	005722	#95-2587 199-7728 199-7745
XOR.LN		021072	#129-4275 399-18104 406-18454
XSALWA	=	0000C0	#6-13
XSALS	=	000040	#5-13
XSOFFS	=	000400	#6-13
XSTRUE	=	000020	#6-13
SEND.L		113564	#561-26535
SPATCH		003452	#7-338
SSAVE2		004540	G 32-1022 37-1269 #39-1344 112-3476 125-4048 135-4563 137-4661 138-4722 140-4821
			141-4882 143-4980 144-5041 146-5140 147-5199 149-5300 150-5361 152-5462 153-5536
			156-5647 157-5723 160-5835 167-6169 169-6283 171-6361 173-6473 175-6565 178-6685
			182-6862 184-6990 188-7189 209-8183 252-10394 261-10802 268-11175 275-11572 328-14331
SSAVE3		004554	G 439-20241 472-21937 550-25956 562-26559
			11-169 12-217 16-384 17-438 27-829 28-880 #39-1351 98-2796 110-3368
			203-7927 219-8657 245-10043 249-10221 295-12585 299-12804 306-13179 452-20894 458-21183
SSAVE4		004572	G #39-1359 115-3604 129-4276 216-8504 223-8889 562-26558
SSAVE5		004612	G 33-1091 37-1269 #39-1368 103-3039 107-3224 212-8334 229-9151 234-9436 240-9764
			288-12248 315-13688 323-14104 333-14602 340-14961 347-15321 354-15681 359-15979 366-16361
			373-16698 379-17029 388-17527 397-18003 404-18353 410-18640 415-18932 422-19295 429-19663
			434-19969 445-20555 477-22198 483-22504 487-22757 493-23033 501-23417 507-23718 513-24035
			520-24369 526-24692 535-25160 556-26267 562-26558
ST1		030724	#209-8183 210-8250
ST10		035640	#249-10221 251-10326
ST11		036302	#252-10394 254-10465
ST12		036524	#256-10552 257-10646
ST13		037102	#261-10802 265-11022
ST14		040244	#268-11175 272-11412
ST15		041476	#275-11572 279-11809
ST16		042730	#282-11914 285-12071
ST17		043572	#288-12248 293-12499
ST18		045060	#295-12585 296-12648
ST19		045274	#299-12804 303-13034
ST2		031150	#212-8334 213-8422
ST20		046444	#306-13179 309-13357
ST21		047444	#315-13688 321-13996
ST22		051212	#323-14104 325-14232
ST23		051740	#328-14331 330-14425
ST24		052276	#333-14602 338-14844
ST25		053472	#340-14961 343-15109
ST26		054304	#347-15321 351-15547
ST27		055502	#354-15681 356-15814
ST28		056260	#359-15979 362-16151
ST29		057244	#366-16361 370-16565
ST3		031474	#216-8504 217-8566
ST30		060356	#373-16698 375-16839
ST31		061152	#379-17029 383-17290
ST32		062550	#388-17527 394-17845
ST33		064306	#397-18003 400-18186
ST34		065320	#404-18353 407-18529
ST35		066302	#410-18640 412-18756

SYMBOL	CROSS REFERENCE	VALUE	REFERENCES
\$T36		066760	#415-18932 419-19143
\$T37		070044	#422-19295 425-19464
\$T38		070772	#429-19663 432-19858
\$T39		072046	#434-19969 436-20082
\$T4		031702	#219-8657 221-8774
\$T40		072474	#439-20241 442-20394
\$T41		073452	#445-20555 448-20744
\$T42		074532	#452-20894 454-21036
\$T43		075340	#458-21183 461-21353
\$T44		076270	#463-21445 464-21507
\$T45		076520	#465-21570 466-21619
\$T46		076704	#468-21729 470-21844
\$T47		077402	#472-21937 473-22005
\$T48		077674	#477-22198 480-22368
\$T49		100654	#483-22504 485-22635
\$T5		032352	#223-8889 225-9012
\$T50		101412	#487-22757 489-22866
\$T51		102030	#493-23033 497-23267
\$T52		103216	#501-23417 504-23573
\$T53		104034	#507-23718 510-23878
\$T54		104674	#513-24035 516-24224
\$T55		105736	#520-24369 523-24531
\$T56		106626	#526-24692 530-24906
\$T57		107756	#535-25160 539-25402
\$T58		111250	#541-25510 543-25606
\$T59		111654	#546-25737 548-25850
\$T6		033022	#229-9151 231-9280
\$T60		112322	#550-25956 552-26053
\$T61		112674	#556-26267 559-26427
\$T7		033512	#234-9436 237-9600
\$T8		034370	#240-9764 244-9975
\$T9		035404	#245-10043 247-10115