

M8061 RL11

RLV11/12

RLV12 DISKLESS
CVRLBB0

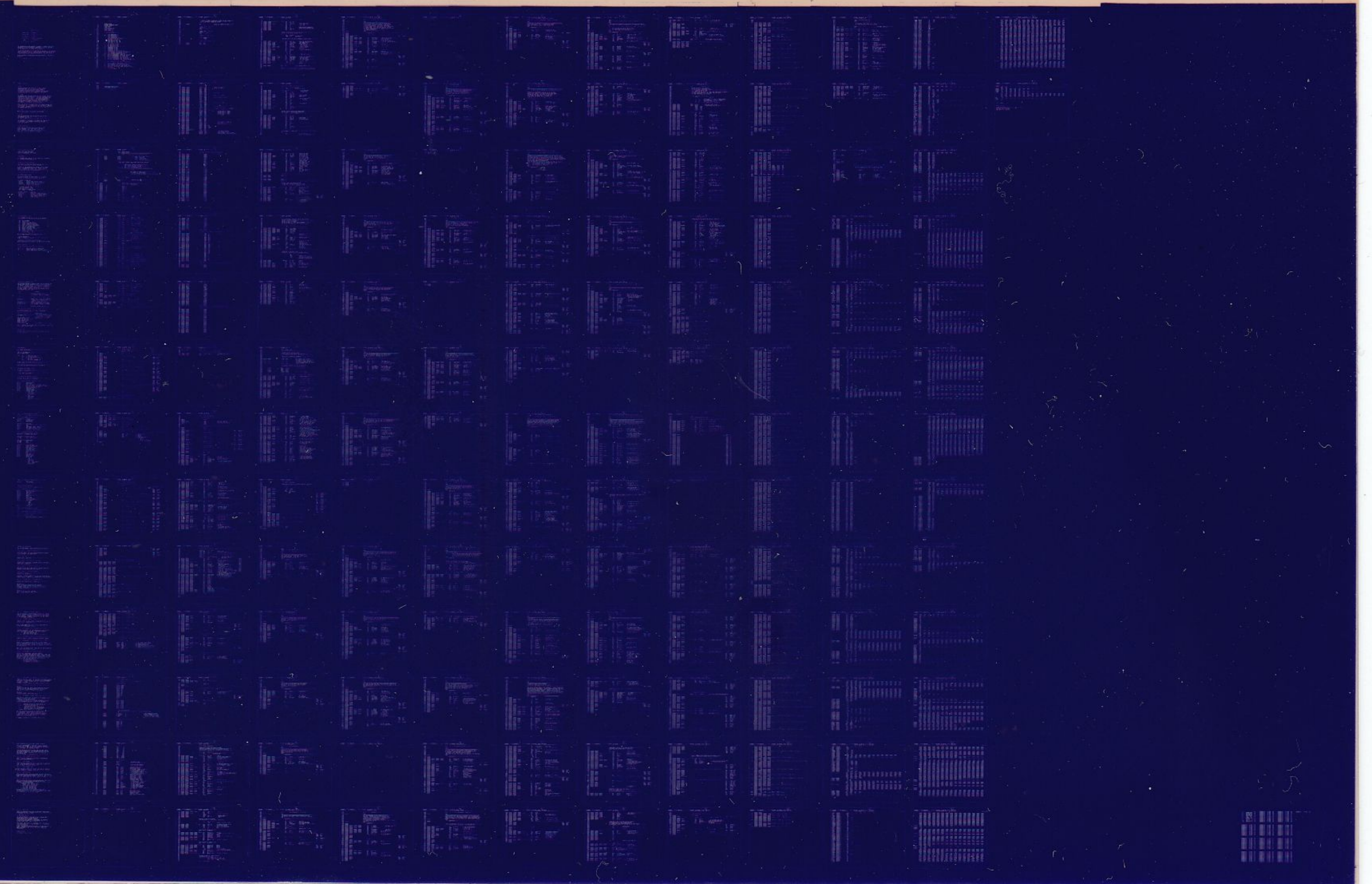
AH-S827B-MC

1 OF 1 JUL 1985

COPYRIGHT© 1982-85

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: AC-S825B-MC
PRODUCT NAME: CVRLB80 RLV12 DISKLESS
DATE CREATED: FEB 1982
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: G. PASQUANTONIO
DATE REVISED: MAR 1985

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

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

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

COPYRIGHT (C) 1982, 1985 DIGITAL EQUIPMENT CORPORATION

GENERAL INFORMATION
-----1.1 ABSTRACT

THE RLV12 DISKLESS TEST IS AN LSI-11 (PDP-11) BASED PROGRAM THAT WILL TEST THE RLV12, RLV11 AND/OR RL11 DISK CONTROLLERS WITH OR WITHOUT DRIVE ATTACHED. THE PROGRAM IS AN ADAPTATION OF "CVRLABO RLV11 RLO1 DSKLS" RETAINING ALL PREVIOUS TEST CAPABILITIES, AND UPGRADED TO INCLUDE ADDITIONAL TESTING IN SUPPORT OF THE RLV12.

RLV12 MODE

THE PROGRAM TESTS THE BASIC INTERFACE LOGIC, CONTROL REGISTER MANIPULATION, AND FUNCTIONALITY. THE RLV12 MAINTENANCE MODE FUNCTION IS EXECUTED TO TEST THE CONTROLLER WRITE/READ DATA PATHS WITHOUT A DRIVE PRESENT. THE EXTENDED ADDRESSING CAPABILITY IS TESTED IN 18 OR 22 BIT MODE DEPENDING ON THE TYPE AND AMOUNT OF MEMORY INSTALLED IN THE TEST SYSTEM. ADDITIONALLY, THE DRIVE BUS INTERFACE LOGIC IS TESTED PROVIDED THE G5388 TEST LOOP MODULE (TLM) IS INSTALLED.

NOTE: THE TLM IS AN IN-HOUSE SPECIAL TEST DEVICE DESIGNED FOR MANUFACTURING USE. IT PROBABLY WILL NOT BE GENERALLY AVAILABLE IN THE FIELD. THE CODE FOR TESTING WITH THE TLM IS INCLUDED IN THIS PROGRAM (BYPASSED BY DEFAULT) TO MAINTAIN CONSISTENCY WITH THE MANUFACTURING VERSION.

RLV11 MODE

SAME AS ABOVE EXCEPT THAT EXTENDED ADDRESSING IS LIMITED TO 18 BITS, AND DRIVE INTERFACE IS NOT TESTED.

RL11 MODE

SINCE THE MAINTENANCE FUNCTION DOES NOT EXIST ON THE RL11, THIS PROGRAM WILL ONLY TEST THE BASIC INTERFACE LOGIC, AND REGISTER FUNCTIONALITY. THE "NOP" COMMAND IS THE ONLY FUNCTION EXECUTED ON THE RL11.

THIS DIAGNOSTIC IS DESIGNED TO RUN UNDER XXDP+ AND REQUIRES THE SERVICES OF THE DIAGNOSTIC SUPERVISOR (DRS REV D). IT WILL RUN STANDALONE UNDER XXDP+ AND IS CHAINABLE UNDER XXDP+, ACT, OR APT.

1.2 HARDWARE REQUIREMENTS

LSI-11 (PDP-11) PROCESSOR WITH 16K OR MORE OF CORE.
MEMORY MANAGEMENT, KT11 OR EQUIVALENT (OPTIONAL).
CONSOLE TERMINAL (LA30, LA36, VT52, VT100, ETC.).
XXDP+ LOAD DEVICE AND DIAGNOSTIC MEDIA (RX01, RX02, ETC.)
RLV12, RLV11, OR RL11 CONTROLLER(S) UNDER TEST (1-8).
G5388 RLV12 TEST LOOP MODULE (1-8, OPTIONAL).

06
1.3

RELATED DOCUMENTS AND STANDARDS

 CHQUSE XXDP+ / SUPR USER MANUAL.
 RLV12 DISK CONTROLLER USERS GUIDE.
 G5388 TLM ENGINEERING SPEC (OPTIONAL).

1.4

MISCELLANEOUS

 ALL HARDWARE OTHER THAN THE UNIT(S) UNDER TEST IS ASSUMED TO
 BE IN PROPER WORKING ORDER. IF NOT (OR YOU DON'T KNOW) RUN
 ALL APPLICABLE SYSTEM DIAGNOSTICS.

2.0

LOADING AND STARTING PROCEDURES

 THIS PROGRAM IS LOADED AND STARTED FROM ANY XXDP+ MEDIA
 USING THE STANDARD XXDP+ OPERATING PROCEDURES.

AT START UP, THE SUPERVISOR WILL IDENTIFY ITSELF AND THE
 NAME OF THIS PROGRAM ON THE CONSOLE TERMINAL, AND THEN
 DISPLAY A COMMAND MODE PROMPT (DR>) WHICH INDICATES
 READY TO ACCEPT ANY OF THE COMMANDS DESCRIBED IN 2.1 BELOW.
 THE GENERALIZED COMMAND STRING FORMAT IS:

DR>COM(MAND)/SWITCH:VALUE/SWITCH:VALUE ... <CR>

2.1

SUPERVISOR COMMAND SUMMARY

 THIS SECTION PRESENTS A BRIEF OVERVIEW OF THE COMMANDS
 NECESSARY TO CONTROL THE OPERATION OF THIS PROGRAM
 UNDER THE XXDP+ DIAGNOSTIC SUPERVISOR (REV D).

THE PRIMARY COMMANDS ARE:

STA(RT)	INITIAL START, BUILD P-TABLES (SEE 2.4).
RES(TART)	RESTART USING EXISTING P-TABLES.
CON(TINUE)	CONTINUE AFTER <+C> OR ERROR HALT.
PRO(CCEED)	PROCEED (FROM ERROR HALT ONLY).
EXI(T)	RETURN TO XXDP+ MONITOR.

THE FOLLOWING SWITCHES APPLY TO THE ABOVE:

/TEST:<TEST NUMBERS TO RUN>
 /FLAG:<SEE FLAG LIST BELOW>
 /EOP:<NUMBER OF PASSES 'TIL END-OF-PASS REPORT>
 /PASS:<NUMBER OF PASSES TO RUN>

ADDITIONAL COMMANDS AVAILABLE ARE:

DRO(P)/UNIT:N	REMOVE UNIT N FROM TEST LIST.
ADD/UNIT:N	ADD UNIT N (PREVIOUSLY DROPPED).
DIS(PLAY)/UNIT:N	PRINT UNITS P-TABLE ENTRIES.
PRI(NT)	PRINT OPTIONAL REPORTS (IF ANY).
FLA(GS)	PRINT CURRENT FLAG SETTINGS.
ZFL(AGS)	CLEAR ALL FLAGS.

RUN TIME OPTIONS (FLAGS)

 THE FOLLOWING FLAGS ARE USED IN LIEU OF THE HARDWARE
 SWITCH REGISTER TO FURTHER DEFINE PROGRAM BEHAVIOUR:

HOE HALT ON ERROR
 LOE LOOP ON ERROR
 IER INHIBIT ALL ERROR REPORTS
 IBR INHIBIT BASIC ERROR REPORTS
 IXR INHIBIT EXTENDED ERROR REPORTS
 PRI DIRECT OUTPUT TO A LINE PRINTER
 PNT PRINT TEST NUMBERS AS EXECUTED
 BOE GOOD OLD "BELL-ON-ERROR"
 UAM RUN IN "UNATTENDED MODE" (UNUSED)
 ISR INHIBIT STATISTICAL REPORTS
 IDR INHIBIT "AUTO-DROP"
 ADR EXECUTE "AUTO-DROP" IF DEVICE NOT READY
 LOT LOOP ON TEST
 EVL EVALUATE ERRORS (UNUSED)

FLAG SETTINGS ARE ALTERED BY USING THE /FLAGS: SWITCH
 IN ANY COMMAND STRING. TWO EXAMPLES FOLLOW:

STA/FLA:IER:BOE<CR>
 START THE PROGRAM, RUN ALL TEST IN ORDER, INHIBIT ERROR
 REPORTS AND RING BELL INSTEAD.

RES/RES:1-5,7,9-29/PAS:10/FLA:HOE:BOE<CR>
 RESTART AND RUN ALL TESTS EXCEPT 6, 8, AND 30 FOR 10 PASSES.
 RING THE BELL AND HALT (RETURN TO COMMAND MODE) ON ANY ERROR.

2.3

CONTROL CHARACTERS

 THE FOLLOWING CONTROL CHARACTERS ARE RECOGNIZED:

CTRL C <+C> ABORT AND RETURN TO COMMAND MODE.
 CTRL O <+O> SUPPRESS TTY OUTPUT UNTIL NEXT <+O>.
 CTRL Z <+Z> DURING START/RESTART PARAMETER ENTRY ONLY.
 IMPLIES TAKE DEFAULTS ON ALL REMAINING QUERIES.

HARD AND SOFT PARAMETER SETUP

 THE FOLLOWING DIALOGUE IS TAKEN AT START TIME TO ESTABLISH THE HARDWARE AND SOFTWARE PARAMETER TABLES (P-TABLES) REQUIRED BY THE PROGRAM. DEFAULT VALUES, WHEN APPLICABLE, APPEAR JUST TO THE LEFT OF THE QUESTION MARK (?). TYPE <CR> TO ACCEPT THE DEFAULT, OR ANSWER THE QUERY APPROPRIATELY. A <+Z> TYPED AT ANY POINT DURING THE SESSION IMPLIES TAKE DEFAULTS FOR ALL REMAINING QUERIES. NOTE THAT ON RESTART OR CONTINUE, ONLY THE SOFTWARE DIALOGUE IS TAKEN.

QUERY	RESPONSE

CHANGE HW (L) ?	NO DEFAULT, ANSWER Y OR N. IF N, ASSUME 1 UNIT IN THE DEFAULT CONFIGURATION, AND GO TO "CHANGE SW".
# UNITS (D) ?	NO DEFAULT, ENTER # OF UNITS TO TEST.
RLV12 (L) Y ?	ANSWER Y OR N. IF N, SKIP NEXT QUERY.
BAE ENABLED (L) Y ?	ON RLV12, BAE IS ENABLED BY DEFAULT... ...ANSWER Y OR N, AND SKIP NEXT QUERY.
RLV11 (L) N ?	ANSWER Y IF RLV11, N IF RL11.
CSR ADDRESS (O) 174400 ?	ENTER RL CSR ADDRESS IF DIFFERENT.
VECTOR (O) 160 ?	ENTER INTERRUPT VECTOR IF DIFFERENT.
BR LEVEL (O) 4 ?	ENTER INTERRUPT PRIORITY IF DIFFERENT.
CHANGE SW (L) ?	NO DEFAULT, ANSWER Y OR N. IF N, BYPASS ALL REMAINING QUERIES.
ERROR LIMIT FOR AUTO-DROP (O) 0 ? SET ERROR LIMIT (IF ANY).	
ALL REMAINING QUERIES ARE FOR OPTIONAL (MANUFACTURING) G5388 TEST-LOOP-MODULE SET-UP. USE <+Z> TO BYPASS.	
G5388 TLM INSTALLED (L) N ?	ANSWER Y OR N. IF NO, BYPASS THE REST.
CSR ADDRESS (UNIT 0) (O) 160010	? BASE TLM CSR (SEE NOTE BELOW).
DUMP PROM ON TTY: (L) N ?	IF Y, GO DIRECTLY TO THE DUMP... ...UTILITY (SEE 7.16.7 BELOW).
PROM ID NUMBER (O) 401 ?	ENTER PROM ID IF DIFFERENT.
INHIBIT MAX PEAK SHIFT (L) N ?	ANSWER Y OR N...
INHIBIT MIN PEAK SHIFT (L) N ?	
INHIBIT NOMINAL CLOCK (L) N ?	
INHIBIT FAST CLOCK (L) N ?	
INHIBIT SLOW CLOCK (L) N ?	
SECTOR NUMBER (O) 0 ?	SECTOR NUMBER TO TEST (0 = ALL).

NOTE: IF RUNNING MULTIPLE UNITS WITH TLM'S (UNLIKELY), THE PROGRAM EXPECTS THAT EACH ADDITIONAL TLM CSR IS OFFSET BY 10 (OCTAL).
 I.E. 160010, 160020, 160030, ETC...

2.5

EXECUTION TIME

 EXECUTION TIME IS DEPENDANT UPON CPU TYPE, MEMORY SIZE, AND BUS TYPE (18 VS 22 BIT), BUT SHOULD NOT EXCEED 30 SECONDS IN ANY CONFIGURATION.

ERROR REPORTING

ALL ERRORS ARE REPORTED ON THE CONSOLE TERMINAL AS THEY OCCUR.
THE GENERAL ERROR FORMAT IS:

CVRLB XXX ERR NNNNN UNIT NN TEST TTT SUB SSS PC: PPPPP
ONE LINE DESCRIPTION.....
EXP'D: 000000 REC'D: 177777

WHERE: XXX IS ERROR TYPE (HRD OR SFT),
NNNNN IS THE ERROR NUMBER,
NN IS THE FAILING UNIT NUMBER,
TTT IS THE TEST...
SSS ...AND SUBTEST NUMBERS, AND
PPPPPP IS THE PC OF THE ERROR CALL.

IN MANY CASES, THE ENTIRE CONTROLLER STATE (ALL REGISTERS
BEFORE AND AFTER THE FACT) IS ALSO DISPLAYED.

ERROR REPORTING AND RECOVERY MAY BE ALTERED AND/OR INHIBITED
VIA THE /FLAG: SWITCH AS DESCRIBED IN 2.2 ABOVE.

4.0 PERFORMANCE AND PROGRESS REPORTS

THE OPERATING ENVIRONMENT IS DISPLAYED ON THE CONSOLE
TERMINAL AT START/RESTART TIME.

A TOTAL (CUMULATIVE) ERROR COUNT IS DISPLAYED AT THE END
OF EVERY PASS, THEREAFTER.

5.0 RLV12 CONTROLLER REGISTER DEFINITION

THE RLV12, RLV11, AND RL11 CONTROLLERS UTILIZE THE FOLLOWING
REGISTERS FOR CONTROL OF THE SUBSYSTEM:

5.1 RLCS -- CONTROL AND STATUS REGISTER (174400)

BIT<15> COMPOSITE ERROR
BIT<14> DRIVE ERROR
BIT<13> NON-EX MEM (OPI=0), OR PARITY (OPI=1)
BIT<12> DATA LATE (OPI=0), OR HEADER NOT FOUND (OPI=1)
BIT<11> DATA CRC (OPI=0), OR HEADER CRC (OPI=1)
BIT<10> OPERATION INCOMPLETE (OPI)
BIT<9:8> DRIVE SELECT (3 TO 0)
BIT<7> CONTROLLER READY (OR DONE)
BIT<6> INTERRUPT ENABLE
BIT<5:4> EXTENDED ADDRESS BITS<17:16>
BIT<3:1> FUNCTION CODE:
0 = MAINT (RLV12/11), NOP (RL11)
1 = WRITE CHECK
2 = GET DRIVE STATUS
3 = SEEK
4 = READ HEADER
5 = WRITE DATA
6 = READ DATA
7 = READ WITHOUT HEADER CHECK
BIT<0> DRIVE READY

06
5.2

RLBA -- BUS ADDRESS REGISTER (174402)

BIT<15:0> BUS ADDRESS FOR DMA DATA EXCHANGE.

5.3

RLDA -- DISK ADDRESS REGISTER (174404)

FOR READ/WRITE FUNCTIONS:BIT<15:7> CYLINDER NUMBER
BIT<6> HEAD SELECT
BIT<5:0> SECTOR NUMBER

FOR SEEK/STATUS FUNCTIONS:

BIT<15:7> SEEK DIFFERENCE (SEEK), UNUSED (STATUS)
BIT<6:5> ZERO
BIT<4> HEAD SELECT (SEEK), ZERO (STATUS).
BIT<3> ZERO (SEEK), RESET (STATUS)
BIT<2> SEEK DIRECTION (SEEK), ZERO (STATUS)
BIT<1> 0 = SEEK, 1 = GET STATUS
BIT<0> MARKER, MUST BE 1 FOR EITHER FUNCTION.

5.4

RLMP -- MULTIPURPOSE REGISTER (174406)

WRITE BEFORE READ/WRITE DATA FUNCTIONS:
BIT<15:0> NEGATIVE WORD COUNT

READ AFTER MAINTENANCE FUNCTION:

1ST WORD: CRC OF INITIAL (DA)+3
2ND WORD: CRC OF CRC OF INITIAL (DA)+4

READ AFTER READ-HEADER FUNCTION:

1ST WORD: DISK ADDRESS.
2ND WORD: ZERO.
3RD WORD: HEADER CRC.

READ AFTER GET STATUS FUNCTION:

BIT<15> WRITE DATA ERROR (WDE)
BIT<14> CURRENT HEAD ERROR (CHE)
BIT<13> WRITE LOCK (WL)
BIT<12> SEEK TIME-OUT (SKTO)
BIT<11> SPIN ERROR (SPE)
BIT<10> WRITE GATE ERROR (WGE)
BIT<9> VOLUME CHECK (VC)
BIT<8> DRIVE SELECT ERROR (DSE)
BIT<7> ZERO
BIT<6> HEAD SELECT (HS)
BIT<5> COVER OPEN (CO)
BIT<4> HEADS OUT (HO)
BIT<3> BRUSHES HOME (BH)
BIT<2:0> DRIVE STATE:
 0 = LOAD
 1 = SPIN UP
 2 = BRUSH CYCLE
 3 = LOAD HEADS
 4 = SEEK - TRACK COUNTING
 5 = SEEK - LINEAR MODE
 6 = UNLOAD HEADS
 7 = SPIN DOWN

06
5.5 RLBAE -- BUS ADDRESS EXTENSION (174410) -- RLV12 ONLY

BIT<15:6> ZERO (UNUSED)
BIT<5:0> EXTENDED ADDRESS BITS<21:16>

6.0 G5388 TLM REGISTER DEFINITION

THE G5388 TLM UTILIZES THE FOLLOWING REGISTERS:

6.1 TCSR -- CONTROL AND STATUS (160010)

BIT<15> NEW SEEK/STATUS COMMAND RECEIVED.
BIT<14> WRITE GATE ERROR.
BIT<13> WRITE GATE.
BIT<12> WRITE DATA (LATCH).
BIT<11> SYS-CLK.
BIT<10:9> DRIVE NUMBER SELECTED.
BIT<8> PWR-OK.
BIT<7> DRIVE READY.
BIT<6> DRIVE ERROR.
BIT<5> PEAK SHIFT ENABLE.
BIT<4> SECTOR GENERATOR ENABLE.
BIT<3:2> CLOCK SELECT:
0 - SLOW
1 - NOMINAL
2 - FAST
3 - PROM READ MODE
BIT<1> CLEAR FLAGS.
BIT<0> TLM RESET.

6.2 TSKGS -- SEEK/STATUS REGISTER (160012)

BIT<15:0> HOLDS DRIVE COMMAND RECEIVED FROM RLV12.

6.3 TPDS -- PSUEDO-DRIVE STATUS REGISTER (160014)

BIT<15:0> SENDS PSUEDO-DRIVE STATUS TO THE RLV12.

6.4 TPROM -- PROM REGISTER (160016)

BIT<15:0> HOLDS PROM ID DURING "TLM RESET" AND/OR
PROM DATA DURING "READ PROM" MODE.

- 06
7.0 **HARDWARE TEST DESCRIPTION**

 THIS SECTION PROVIDES A BRIEF DESCRIPTION OF THE HARDWARE TESTS.
 REFER TO THE PROGRAM LISTING (8.0) FOR FURTHER DETAILS.
- 7.1 **TESTS 1 THRU 5 -- REGISTER ADDRESSABILITY.**

 THESE TESTS VERIFY THAT EACH CONTROLLER REGISTER ANSWERS TO
 IT'S BUS ADDRESS. BOTH READ AND WRITE ACCESS IS VERIFIED
 USING "MOV" AND "TST" INSTRUCTIONS.
- 7.2 **TEST 6 -- BUS RESET.**

 VERIFIES THAT A "BUS-RESET" PLACES THE CONTROLLER IF THE
 PROPER "INITIALIZED" STATE.
- 7.3 **TESTS 7 THRU 10 -- READ AND WRITE REGISTERS.**

 VERIFIES THAT WE CAN WRITE TO AND READ FROM ALL REGISTERS
 (EXCEPT MPR). VARIOUS DATA PATTERNS ARE WRITTEN AND VERIFIED
 USING "MOV" INSTRUCTIONS.
- 7.4 **TESTS 11 THRU 14 -- BIS AND BIC REGISTERS.**

 VERIFIES THAT WE CAN SET AND CLEAR ALL WRITEABLE BITS IN ALL
 REGISTERS (EXCEPT MPR) USING "BIS" AND "BIC" INSTRUCTIONS.
 VARIOUS DATA PATTERNS ARE EMPLOYED HERE TOO.
- 7.5 **TESTS 15 THRU 19 -- REGISTER UNIQUENESS.**

 VERIFIES THAT WRITING TO ANY REGISTER HAS NO AFFECT ON ANY
 OTHER (NO DUAL-ADDRESSING).
- EXCEPTION: EXTENDED ADDRESS BITS<17:16> MAY BE WRITTEN VIA
 EITHER RLCS<5:4> OR RLBAE<1:0>. REGARDLESS OF WHICH REGISTER IS
 ACTUALLY WRITTEN, THOSE BITS ARE COPIED INTO THE OTHER REGISTER.
- 7.6 **TEST 20 -- FUNCTION CODE 0, MAINTENANCE OR NOP.**

 VERIFIES THAT FUNCTION CODE 0 EXECUTES AS A "MAINTENANCE"
 FUNCTION (RLV12 AND RLV11), OR "NOP" (RL11).
- FOR RLV12/11:
 EXPECT TO EXECUTE A MAINTENANCE MODE SEQUENCE.
 CHECK THAT THE MAINTENANCE FUNCTION COMPLETES WITH NO
 CONTROLLER ERRORS, AND THAT THE FINAL VALUE IN THE RLDA
 IS CORRECT, INDICATING THAT THE INTERNAL MAINTENANCE
 MODE SEQUENCER FINISHED IT'S CYCLE.
- FOR RL11:
 EXPECT TO EXECUTE A "NOP" FUNCTION.
 CHECK THAT ALL ERROR BITS ARE CLEAR (RLCS<13:10>)
 AND ALL OTHER REGISTERS UNAFFECTED.

- 06
7.7 TEST 21 -- INTERRUPT ON FUNCTION COMPLETE.

THIS TEST EXECUTES A NOP/MAINTENANCE FUNCTION WITH INTERRUPT ENABLED, AND VERIFIES THAT AN INTERRUPT THRU THE RL VECTOR ACTUALLY OCCURS. INCORRECT VECTORS UNDER 1000 ARE TRAPPED BY THE SUPERVISOR. INCORRECT VECTORS TO ANY OTHER ADDRESS WILL YIELD UNPREDICTABLE RESULTS.
- 7.8 TEST 22 -- INTERRUPT PRIORITY LEVEL TEST.

SIMILAR TO THE LAST TEST, EXCEPT THAT WE REITERATE AT ALL CPU LEVELS (7 TO 0), VERIFYING THAT THE INTERRUPT IS HELD OFF WHEN THE CPU LEVEL => CONTROLLER LEVEL.

NOTE: IF UNIT UNDER TEST IS RL11, YOU'LL SEE AN END-PASS AT THE COMPLETION OF THIS TEST. ALL FOLLOWING TESTS UTILIZE THE RLV12/11 MAINTENANCE FUNCTION.
- 7.9 TEST 23 -- MAINTENANCE, FORCE OPI ERROR (RLV12/11 ONLY).

THIS TEST WILL VERIFY THAT THE CONTROLLER WILL SET THE OPI (OPERATION INCOMPLETE) BIT ON A MAINTENANCE FUNCTION IF THE INITIAL WORD COUNT IS INCORRECT. WE'LL DO 2 PASSES.
1. WORD COUNT LESS THAN 510.
2. WORD COUNT GREATER THAN 511.
BOTH CASES SHOULD CAUSE AN OPI ERROR IN THE CONTROLLER.
- 7.10 TEST 24 -- MAINTENANCE, FORCE OPI AND INTERRUPT (RLV12/11 ONLY).

SIMILAR TO THE LAST, EXCEPT EXECUTE WITH INTERRUPT ENABLED. EXPECT OPI ERROR TO CAUSE AN INTERRUPT THRU THE RL VECTOR.
- 7.11 TEST 25 -- MAINTENANCE, OPI TIMING TEST (RLV12/11 ON LSI ONLY).

VERIFY THAT THE SETTING OF THE OPI BIT IS TIMED CORRECTLY. EXECUTE A MAINTENANCE FUNCTION IN FLAG MODE WITH INVALID WORD COUNT TO FORCE OPI ERROR. MEASURE THE TIME FROM "GO" UNTIL THE OPI ERROR FLAG SETS AND CHECK THAT TIME AGAINST THE SPEC LIMITS (155 TO 650 MSECS.).

NOTE: THIS TEST EMPLOYS A SOFT TIMER THAT HAS BEEN CALIBRATED FOR LSI11/2,/23,/73 CPUS ONLY. IF YOUR CPU IS NOT ONE OF THESE THIS TEST IS AUTOMATICALLY BYPASSED.
- 7.12 TEST 26 -- MAINTENANCE, FIFO AND DMA TRANSFER (RLV12/11 ONLY).

VERIFIES THAT "MAINTENANCE" FUNCTIONS CORRECTLY. EXECUTE A MAINTENANCE MODE FUNCTION IN FLAG MODE. VERIFY THE FINAL BA, DA, AND BAE REGISTERS ARE CORRECT. CHECK THE SERIAL WRITE/READ DATA PATHS BY READING THE TWO MAINTENANCE CRC WORDS (CRC OF DA+3, AND CRC OF CRC OF DA+4) FROM THE FIFO VIA THE MP REGISTER. VERIFY THAT 256 WORDS WERE TRANSFERED FROM MEMORY TO THE FIFO, AND 255 WORDS FROM FIFO BACK TO MEMORY. REPEAT 57 TIMES USING VARIOUS DATA.
1. 28 DIFFERENT DATA PATTERNS.
2. THE COMPLIMENT OF THOSE PATTERNS.
3. ONE RANDOM 256 WORD PATTERN.

- 06
7.13 TEST 27 -- MAINTENANCE, FIFO ADDRESS TEST (RLV12/11 ONLY).

SIMILAR TO THE LAST, EXCEPT THAT THE DATA IS AN ADDRESS PATTERN WHERE EACH FIFO LOCATION WILL BE WRITTEN WITH IT'S OWN ADDRESS (0 TO 255.). REPEAT A SECOND TIME WITH A COMPLIMENT ADDRESS PATTERN (-1 TO -256.). INTERRUPT ON DONE IS ENABLED AND EXPECTED DURING THIS TEST.
- 7.14 TEST 28 -- MAINTENANCE, BANK SELECT 7 AND NXM (RLV12/11 ONLY).

FOR RLV12:
VERIFIES THAT BBS7 WILL SELECT THE I/O PAGE AND THAT ACCESS TO LOCATION 0 IN THAT PAGE WILL GENERATE NXM AND OPI ERRORS.
NOTE: IF BANK 7 IS NOT PROPERLY SELECTED, 1000 BYTES STARTING AT XXXX1000 WILL PROBABLY GET CRUNCHED !!!
- FOR RLV11:
SINCE RLV11 DOESN'T ASSERT BBS7, WE'LL EXECUTE THIS TEST ONLY IF MEMORY SIZE IS LESS THAN 124K WORDS.
- 7.15 TEST 29 -- MAINTENANCE, EXTENDED MEMORY ACCESS (RLV12/11 ONLY).

THIS TEST WILL VERIFY THAT THE CONTROLLER CAN ADDRESS EXTENDED MEMORY UP TO 124K (18 BIT) OR 2044K (22 BIT) DEPENDING UPON CONTROLLER TYPE AND CONFIGURATION.
IT REQUIRES THE SERVICES OF THE MEMORY MANAGEMENT FACILITY (KT11 OR EQUIVALENT IN 11/23) TO ESTABLISH AND VERIFY DATA BUFFERS IN MEMORY FROM 32K TO 2044K, IN 4K INCREMENTS.
1. SETUP -- USING MMU, INITIALIZE A WRITE BUFFER WITH A RANDOM DATA PATTERN, AND CLEAR THE READ BUFFER.
 2. EXECUTE -- RLV12/11 MAINTENANCE MODE.
WRITE BUFFER => FIFO => READ BUFFER.
 3. VERIFY -- USING MMU, VERIFY THAT THE READ BUFFER RECEIVED A COPY OF THE DATA IN THE WRITE BUFFER.
 4. INCREMENT BUFFER ADDRESS BY 4K, AND REPEAT.
- NOTE THAT THE ENTIRE 18 (OR 22) BIT ADDRESS SPACE IS TESTED ON 4K BOUNDARIES, WHETHER MEMORY EXISTS THERE OR NOT. IF MEMORY EXISTS, VERIFY THE DATA EXCHANGE AS DESCRIBED. IF NOT, VERIFY THAT THE CONTROLLER GOT A "NXM ERROR". CONTINUE ALL THE ABOVE UNTIL THE I/O PAGE IS REACHED AT 760000 (18 BIT) OR 17760000 (22 BIT).
- OF COURSE, IF AN MMU ISN'T AVAILABLE, THEN ALL THIS IS ACADEMIC, AND WE'LL JUST FALL THRU TO THE NEXT TEST.

DRIVE INTERFACE TESTS (RLV12 ONLY, G5388 TLM REQ'D).

THIS IS AN ASSORTMENT OF 6 TESTS AND 1 UTILITY DESIGNED TO VERIFY THE INTEGRITY OF THE DRIVE INTERFACE SECTION OF THE RLV12 CONTROLLER. ALL REQUIRE THE SERVICES OF THE SPECIAL TEST-LOOP-MODULE (TLM) DESIGNATED G5388.

FROM THE CONTROLLERS POINT OF VIEW, THE TLM LOOKS LIKE A 6 SECTOR RLO1/RLO2 DISK. VARIOUS HEADER AND DATA PATTERNS ARE STORED IN PROM ON THE TLM, AND ARE ACCESSABLE THRU THE EXECUTION OF "NORMAL" READ/WRITE AND STATUS FUNCTIONS IN THE RLV12 CONTROLLER.
REFER TO THE G5388 TLM ENGINEERING SPEC FOR FURTHER DETAILS.

7.16.1 TEST 30 -- STATIC CONTROL AND STATUS BITS.

CHECK THE STATIC CONTROL AND STATUS BITS TO AND FROM THE RLV12. THESE BITS INCLUDE SYS-CLK, PWR-OK, DRIVE-ERROR, DRIVE-READY, AND DRIVE-SELECTS.

7.16.2 TEST 31 -- DRIVE COMMAND, STATUS.

VERIFY THE DRIVE COMMAND AND STATUS CLOCK LOGIC BY EXECUTING A GET-STATUS COMMAND IN THE RLV12. CHECK THAT A DUMMY STATUS WORD IS CORRECTLY RECEIVED FROM THE TLM.

7.16.3 TEST 32 -- DRIVE COMMAND, SEEK.

VERIFY THE DRIVE COMMAND AND SECTOR PULSE LOGIC BY EXECUTING A SEEK COMMAND IN THE RLV12. CHECK THAT THE SEEK DIFFERENCE WORD WAS CORRECTLY TRANSMITTED TO THE TLM.

7.16.4 TEST 33 -- WRITE DATA PATH.

VERIFY THE WRITE GATE, WRITE GATE ERROR, AND WRITE DATA PATH BY EXECUTING A WRITE-DATA FUNCTION IN THE RLV12. THIS TEST ONLY PROVES THAT TRANSITIONS ARE TAKEING PLACE ON THE WRITE DATA LINE SINCE THE PSUEDO-SECTORS ARE ACTUALLY READ-ONLY (PROM).

7.16.5 TEST 34 -- READ DATA PATHS.

VERIFY THE READ DATA PATHS BY EXECUTING READ-DATA, READ-HEADER, AND READ-DATA-WITHOUT-HEADER FUNCTIONS IN THE RLV12. BY DEFAULT, EACH PSUEDO-SECTOR IS EXERCISED SIX TIMES, USING THE FOLLOWING CLOCK VARIATIONS:

1. NOMINAL CLOCK, MIN PEAK SHIFT.
2. NOMINAL CLOCK, MAX PEAK SHIFT.
3. FAST CLOCK, MIN PEAK SHIFT.
4. FAST CLOCK, MAX PEAK SHIFT.
5. SLOW CLOCK, MIN PEAK SHIFT.
6. SLOW CLOCK, MAX PEAK SHIFT.

RETURNED DATA AND STATUS ARE VERIFIED IN EVERY CASE. PEAK SHIFT AND CLOCKS MAY BE SELECTIVELY INHIBITED AND A SINGLE SECTOR EXERCISED (VS ALL OF THEM) AS DESCRIBED IN THE P-TABLE SOFT PARAMETER SETUP (SECTION 2.4) ABOVE.

7.16.6 TEST 35 -- WRITE CHECK.

VERIFY THE READ/WRITE DATA PATHS BY EXECUTING A WRITE-CHECK
FUNCTION IN THE RLV12. EXECUTE USEING CLOCK AND SECTOR OPTIONS
AS DESCRIBED FOR TEST 34 ABOVE.

7.16.7 TLM PROM DUMP UTILITY.

THE FINAL SECTION IS A "PROM DUMP UTILITY" ROUTINE WHICH
CAN BE USED TO PRINT THE CONTENTS OF ANY OF THE 6 TLM
PSUEDO-SECTORS ON THE CONSOLE TERMINAL.
THIS ROUTINE IS NOT INCLUDED IN THE NORMAL TEST SEQUENCE.
IT MUST BE CALLED DIRECTLY AT START/RESTART TIME VIA THE
SOFTWARE CHANGE QUERY "DUMP TLM PROM ON TTY: ?"

WHEN CALLED, THE PROGRAM WILL INTRODUCE ITSELF, AND ASK FOR
A TLM UNIT NUMBER (0 - 7), SECTOR NUMBER (1 - 6), AND DUMP
FORMAT (OCTAL OR HEX).
IT WILL THEN READ THAT SECTOR AND PRINT IT IN THE SPECIFIED
FORMAT, 8 WORDS PER LINE FOR 20 LINES (160. WORDS TOTAL).
WHEN YOU'RE DONE, TYPE <↑C> TO EXIT BACK TO SUPERVISOR
COMMAND MODE.

8.0 PROGRAM LISTING

THE PROGRAM LISTING FOLLOWS:

TABLE OF CONTENTS

14	PROGRAM HEADER
59	HARDWARE PARAMETER CODING
101	SOFTWARE PARAMETER CODING
153	GLOBAL EQUATES
229	GLOBAL DATA
461	INITIALIZATION CODE
658	GLOBAL SUBROUTINES
1015	MEMORY SIZER
1125	REPORT ENVIRONMENT
1146	
1147	RL DISKLESS CONTROLLER TESTS.
1148	
1149	1 -- RLCS ADDRESSABILITY.
1171	2 -- RLBA ADDRESSABILITY.
1193	3 -- RLDA ADDRESSABILITY.
1215	4 -- RLMP ADDRESSABILITY.
1237	5 -- RLBAE ADDRESSABILITY (RLV12 ONLY).
1266	6 -- BUS RESET OF ALL REGISTERS.
1315	7 -- READ WRITE OF RLCS.
1344	8 -- READ WRITE OF RLBA.
1371	9 -- READ WRITE OF RLDA.
1394	10 -- READ WRITE OF RLBAE (RLV12 ONLY).
1421	11 -- BIS AND BIC OF RLCS.
1464	12 -- BIS AND BIC OF RLBA.
1501	13 -- BIS AND BIC OF RLDA.
1534	14 -- BIS AND BIC OF RLBAE (RLV12 ONLY).
1572	15 -- UNIQUENESS OF RLCS.
1611	16 -- UNIQUENESS OF RLBA.
1652	17 -- UNIQUENESS OF RLDA.
1693	18 -- UNIQUENESS OF RLMP.
1741	19 -- UNIQUENESS OF RLBAE (RLV12 ONLY).
1777	20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).
1831	21 -- TEST INTERRUPT ON FUNCTION (0) COMPLETE.
1856	22 -- TEST INTERRUPT PRIORITY BR LEVEL.
1894	23 -- RLV11/12 MAINTENANCE, FORCED OPI (WC <> 511.)
1932	24 -- RLV11/12 MAINTENANCE, FORCED OPI INTERRUPT.
1954	25 -- RLV11/12 MAINTENANCE, OPI TIMING TEST.
1990	26 -- RLV11/12 MAINTENANCE, FIFO DMA AND CRC CHECK.
2123	27 -- RLV11/12 MAINTENANCE, FIFO ADDRESSING.
2215	28 -- RLV11/12 MAINTENANCE, BANK 7 SELECT AND NEXM TEST.
2261	29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.
2386	
2387	RLV12 DRIVE INTERFACE TESTS (G5388 TLM REQUIRED).
2388	
2392	30 -- SYS CLK, PWR OK, DRIVE SELECT, READY, AND ERROR BITS.
2453	31 -- DRIVE COMMAND, STATUS AND STATUS CLOCK.
2490	32 -- DRIVE COMMAND, SEEK DIFF AND SECTOR PULSE.
2521	33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.
2568	34 -- READ DATA, READ HEADER, AND READ DATA W/O HEADER.
2697	35 -- WRITE CHECK.
2895	TLM PROM DUMP UTILITY.

CVRLBBO -- RLV12 DISKLESS. MACY11 30(1046) 12-JUN-85 15:42
CVRLBB.P11 12-JUN-85 15:33 TABLE OF CONTENTS

SEQ 0015

3102	
3103	SUPERVISOR DISPATCH TABLE.
3123	GLOBAL ERROR HANDLERS AND ASCII TEXT
3374	DEVELOPMENT/DEBUG AIDS
3375	
3376	RLV12 EMULATOR

(5)	002020	L\$SPCP::			; POINTER TO S.W. QUES.
(4)	002020	002412	.WORD	L\$SOFT	
(5)	002022	L\$HPTP::			; PTR. TO DEF. H.W. PTABLE
(4)	002022	002376	.WORD	L\$HW	
(5)	002024	L\$SPTP::			; PTR. TO S.W. PTABLE
(4)	002024	003304	.WORD	L\$SW	
(5)	002026	L\$LADP::			; DIAG. END ADDRESS
(4)	002026	040416	.WORD	L\$LAST	
(5)	002030	L\$STA::			; RESERVED FOR APT STATS
(4)	002030	000000	.WORD	0	
(5)	002032	L\$CO::			
(4)	002032	000000	.WORD	0	
(5)	002034	L\$DTYP::			; DIAGNOSTIC TYPE
(4)	002034	000000	.WORD	0	
(5)	002036	L\$APT::			; APT EXPANSION
(4)	002036	000000	.WORD	0	
(5)	002040	L\$DTP::			; PTR. TO DISPATCH TABLE
(4)	002040	026124	.WORD	L\$DISPATCH	
(5)	002042	L\$PRIO::			; DIAGNOSTIC RUN PRIORITY
(4)	002042	000340	.WORD	340	
(5)	002044	L\$ENVI::			; FLAGS DESCRIBE HOW IT WAS SETUP
(4)	002044	000000	.WORD	0	
(5)	002046	L\$EXP1::			; EXPANSION WORD
(4)	002046	000000	.WORD	0	
(5)	002050	L\$MREV::			; SVC REV AND EDIT #
(4)	002050	003	.BYTE	C\$REVISION	
(3)	002051	003	.BYTE	C\$EDIT	
(5)	002052	L\$EF::			; DIAG. EVENT FLAGS
(4)	002052	000000	.WORD	0	
(5)	002054	000000	.WORD	0	
(5)	002056	L\$SPC::			
(4)	002056	000000	.WORD	0	
(5)	002060	L\$DEVP::			; POINTER TO DEVICE TYPE LIST
(4)	002060	002142	.WORD	L\$DVTYP	
(5)	002062	L\$REPP::			; PTR. TO REPORT CODE
(4)	002062	011722	.WORD	L\$RPT	
(5)	002064	L\$EXP4::			
(4)	002064	000000	.WORD	0	
(5)	002066	L\$EXP5::			
(4)	002066	000000	.WORD	0	
(5)	002070	L\$AUT::			; PTR. TO ADD UNIT CODE
(4)	002070	007476	.WORD	L\$AU	
(5)	002072	L\$DUT::			; PTR. TO DROP UNIT CODE
(4)	002072	007414	.WORD	L\$DU	
(5)	002074	L\$LUN::			; LUN FOR EXERCISERS TO FILL
(4)	002074	000000	.WORD	0	
(5)	002076	L\$DESP::			; POINTER TO DIAG. DESCRIPTION
(4)	002076	002122	.WORD	L\$DESC	
(5)	002100	L\$LOAD::			; GENERATE SPECIAL AUTOLOAD EMT
(4)	002100	104035	EMT	E\$LOAD	
(5)	002102	L\$ETP::			; POINTER TO ERR_TBL
(4)	002102	000000	.WORD	0	
(5)	002104	L\$ICP::			; PTR. TO INIT CODE
(4)	002104	006376	.WORD	L\$INIT	
(5)	002106	L\$CCP::			; PTR. TO CLEAN-UP CODE
(4)	002106	007336	.WORD	L\$CLEAN	

```

(5) 002110 L$ACP:: ;PTR. TO AUTO CODE
(4) 002110 007260 .WORD L$AUTO
(5) 002112 L$PRT:: ;PTR. TO PROTECT TABLE
(4) 002112 006370 .WORD L$PROT
(5) 002114 L$TEST:: ;TEST NUMBER
(4) 002114 000000 .WORD 0
(5) 002116 L$DLY:: ;DELAY COUNT
(4) 002116 000000 .WORD 0
(5) 002120 L$HIME:: ;PTR. TO HIGH MEM
(4) 002120 000000 .WORD 0
51
52 002122 DESCRIPT <RLV12 DISKLESS>
(4) 002122 L$DESC:: .ASCIZ /RLV12 DISKLESS/
(3) 002122 046122 030526 020062
(3) 002130 044504 045523 042514
(3) 002136 051523 000
(2) 002142 .EVEN
53
54 002142 DEVTYP <RLV12, RLV11, OR RL11>
(4) 002142 L$DVTYP:: .ASCIZ /RLV12, RLV11, OR RL11/
(3) 002142 046122 030526 026062
(3) 002150 051040 053114 030461
(3) 002156 020054 051117 051040
(3) 002164 030514 000061
(2) .EVEN
55
56 000001 SVCGBL= 1 ; SHOVE EVERYTHING BACK TO THE RIGHT.
57 000001 SVCINS= 1

```



```

59      .SBTTL  HARDWARE PARAMETER CODING
60      ;
61      ; GET PARAMETERS FROM OPERATOR.
62      ;
63      BGNHRD
64      (3) 002170 000036
65      (3) 002172
66      (4) 002172 004130
67      (4) 002174 002266
68      (4) 002176 000002
69      (5) 002200 005044
70      (6) 002202
71      (4) 002202 004130
72      (4) 002204 002276
73      (4) 002206 000001
74      (6) 002210
75      (5) 002210 004004
76      (4) 002212 004130
77      (4) 002214 002313
78      (4) 002216 000001
79      (4) 002220 000031
80      (4) 002222 002323
81      (4) 002224 160000
82      (4) 002226 177776
83      (4) 002230 001031
84      (4) 002232 002340
85      (4) 002234 000000
86      (4) 002236 000776
87      (4) 002240 002032
88      (4) 002242 002351
89      (4) 002244 000340
90      (4) 002246 000000
91      (4) 002250 000007
92      (5) 002252 006004
93      (4) 002254 003032
94      (4) 002256 002363
95      (4) 002260 003400
96      (4) 002262 000000
97      (4) 002264 000007
98      (2) 002266
99      (3) 002266
100     75      ;
101     76      ; THESE ARE THE HARDWARE QUERIES...
102     77      ;
103     78      RL1:  .ASCIZ 'RLV12
104     79      RL1A: .ASCIZ 'BAE ENABLED

```

.WORD L10000-L\$HARD/2
L\$HARD::

.WORD T\$CODE
.WORD RL1
.WORD BIT1

.WORD T\$CODE

.WORD T\$CODE
.WORD RL1A
.WORD BITO

.WORD T\$CODE

.WORD T\$CODE
.WORD RL2
.WORD BITO

.WORD T\$CODE
.WORD CSRA
.WORD T\$LLOLIM
.WORD T\$HILIM

.WORD T\$CODE
.WORD VECA
.WORD T\$LLOLIM
.WORD T\$HILIM

.WORD T\$CODE
.WORD BRL
.WORD 340
.WORD T\$LLOLIM
.WORD T\$HILIM

.WORD T\$CODE
.WORD DRN
.WORD 3400
.WORD T\$LLOLIM
.WORD T\$HILIM

.EVEN
L10000:

```

002304 041101 042514 004504
002312 000
80 002313 122 053114 030461 RL2: .ASCIZ 'RLV11 '
002320 004411 000
81 002323 103 051123 040440 CSRA: .ASCIZ 'CSR ADDRESS '
002330 042104 042522 051523
002336 000011
82 002340 042526 052103 051117 VECA: .ASCIZ 'VECTOR '
002346 004411 000
83 002351 102 020122 042514 BRL: .ASCIZ 'BR LEVEL '
002356 042526 004514 000
84 002363 104 044522 042526 DRN: .ASCIZ 'DRIVE ' ; UNUSED.
002370 004411 000
85 002374 .EVEN
86
87
88

```

;...AND THESE ARE THE SINGLE UNIT DEFAULTS (HARD P-TABLE).

```

89 002374
(3) 002374 000005
(3) 002376
90 002376 174400
91 002400 000160
92 002402 000200
93 002404 000000
94 002406 000003
95
96
97
98
99 002410
(3) 002410

```

BGNHW

```

.WORD 174400
.WORD 160
.WORD PRI04
.WORD 0
.WORD 3

```

```

; (0) CSR BUS ADDRESS.
; (2) VECTOR
; (4) PRIORITY
; (6) DRIVE (BITS 8,9,10)
; (10) CONTROLLER TYPE...
;.....0 = RL11...
;.....1 = RLV11...
;.....2 = RLV12, BAE DISABLED...
;.....3 = RLV12, BAE ENABLED.

```

ENDHW

```

.L$HW:: .WORD L10001-L$HW/2
L10001:

```

```

101      .SBTTL  SOFTWARE PARAMETER CODING
102      ;
103      ; GET SOFT PARAMETERS FROM OPERATOR.
104      ;
105      002410      BGNSFT
(3) 002410 000054      .WORD L10002-L$SOFT/2
(3) 002412      L$SOFT::
106      002412      GPRMD  ADLIM,0,D,-1,0,-1,YES  ; GET AUTO-DROP ERROR LIMIT.
(4) 002412 000052      .WORD T$CODE
(4) 002414 002542      .WORD ADLIM
(4) 002416 177777      .WORD -1
(4) 002420 000000      .WORD T$LLOLIM
(4) 002422 177777      .WORD T$HILIM
107      002424      DISPLAY TLMBP      ; BYPASS MESSAGE.
(4) 002424 000003      .WORD T$CODE
(4) 002426 002575      .WORD TLMBP
108      002430      GPRML  TLMIN,12,BIT0,YES      ; TLM ON THIS UNIT ??
(4) 002430 005130      .WORD T$CODE
(4) 002432 002751      .WORD TLMIN
(4) 002434 000001      .WORD BIT0
109      002436      XFERF  1$
(5) 002436 042044      .WORD T$CODE
110      002440      GPRMA  TCSU0,14,0,160000,177776,YES ; TLM CSR FOR UNIT 0.
(4) 002440 006031      .WORD T$CODE
(4) 002442 002776      .WORD TCSU0
(4) 002444 160000      .WORD T$LLOLIM
(4) 002446 177776      .WORD T$HILIM
111      002450      GPRML  PRMDMP,2,BIT0,YES      ; PROM DUMP ??
(4) 002450 001130      .WORD T$CODE
(4) 002452 003024      .WORD PRMDMP
(4) 002454 000001      .WORD BIT0
112      002456      XFERT  1$
(5) 002456 032024      .WORD T$CODE
113      002460      GPRMD  PRM,4,0,-1,0,-1,YES      ; PROM ID.
(4) 002460 002032      .WORD T$CODE
(4) 002462 003047      .WORD PRM
(4) 002464 177777      .WORD -1
(4) 002466 000000      .WORD T$LLOLIM
(4) 002470 177777      .WORD T$HILIM
114      002472      GPRML  INHMP,6,BIT15,YES      ; INHIBIT MAX PEAK...
(4) 002472 003130      .WORD T$CODE
(4) 002474 003070      .WORD INHMP
(4) 002476 100000      .WORD BIT15
115      002500      GPRML  INHMNP,6,BIT14,YES      ; ...MIN PEAK...
(4) 002500 003130      .WORD T$CODE
(4) 002502 003120      .WORD INHMNP
(4) 002504 040000      .WORD BIT14
116      002506      GPRML  INHNC,6,BIT2,YES      ; ...NOMINAL CLOCK...
(4) 002506 003130      .WORD T$CODE
(4) 002510 003150      .WORD INHNC
(4) 002512 000004      .WORD BIT2
117      002514      GPRML  INHFC,6,BIT1,YES      ; ...FAST CLOCK...
(4) 002514 003130      .WORD T$CODE
(4) 002516 003177      .WORD INHFC
(4) 002520 000002      .WORD BIT1
118      002522      GPRML  INHSC,6,BIT0,YES      ; ...SLOW CLOCK.

```


(4)	002522	003130				.WORD	T#CODE
(4)	002524	003223				.WORD	INMSC
(4)	002526	000001				.WORD	BITO
119	002530			GPRMD	SEC1,10,0,7,0,6,YES		; SINGLE SECTOR OPTION.
(4)	002530	004032				.WORD	T#CODE
(4)	002532	003247				.WORD	SEC1
(4)	002534	000007				.WORD	7
(4)	002536	000000				.WORD	T#LOLIM
(4)	002540	000006				.WORD	T#HILIM
120	002542			1\$:	ENDSFT		
(2)							.EVEN
(3)	002542						L10002:
121							
122							; THESE ARE THE SOFTWARE QUERIES...
123							
124	002542	051105	047522	020122	ADLIM:	.ASCIZ	'ERROR LIMIT FOR AUTO-DROP '
	002550	044514	044515	020124			
	002556	047506	020122	052501			
	002564	047524	042055	047522			
	002572	020120	000				
125							
126	002575	101	046114	051040	TLMBP:	.ASCIZ	'ALL REMAINING QUERIES ARE FOR OPTIONAL (MANUFACTURING)'
	002602	046505	044501	044516			
	002610	043516	050440	042525			
	002616	044522	051505	040440			
	002624	042522	043040	051117			
	002632	047440	052120	047511			
	002640	040516	020114	046450			
	002646	047101	043125	041501			
	002654	052524	044522	043516			
	002662	000051					
127	002664	032507	034063	020070		.ASCIZ	'G5388 TEST-LOOP-MODULE SET-UP. USE <+Z> TO BYPASS.'
	002672	042524	052123	046055			
	002700	047517	026520	047515			
	002706	052504	042514	051440			
	002714	052105	052455	027120			
	002722	020040	051525	020105			
	002730	057074	037132	052040			
	002736	020117	054502	040520			
	002744	051523	000056				
128	002750	000				.BYTE	0
129	002751	107	031465	034070	TLMIN:	.ASCIZ	'G5388 TLM INSTALLED '
	002756	052040	046514	044440			
	002764	051516	040524	046114			
	002772	042105	000011				
130	002776	051503	020122	042101	TCSUO:	.ASCIZ	'CSR ADDRESS (UNIT 0) '
	003004	051104	051505	020123			
	003012	052450	044516	020124			
	003020	024460	000011				
131	003024	052504	050115	050040	PRMDMP:	.ASCIZ	'DUMP PROM ON TTY: '
	003032	047522	020115	047117			
	003040	052040	054524	004472			
	003046	000					
132	003047	120	047522	020115	PRM:	.ASCIZ	'PROM ID NUMBER '
	003054	042111	047040	046525			
	003062	042502	004522	000011			

```

133 003070 047111 044510 044502 INHMXP: .ASCIZ 'INHIBIT MAX PEAK SHIFT '
      003076 020124 040515 020130
      003104 042520 045501 051440
      003112 044510 052106 000011
134 003120 047111 044510 044502 INHMNP: .ASCIZ 'INHIBIT MIN PEAK SHIFT '
      003126 020124 044515 020116
      003134 042520 045501 051440
      003142 044510 052106 000011
135 003150 047111 044510 044502 INHNC: .ASCIZ 'INHIBIT NOMINAL CLOCK '
      003156 020124 047516 044515
      003164 040516 020114 046103
      003172 041517 004513 000
136 003177 111 044116 041111 INHFC: .ASCIZ 'INHIBIT FAST CLOCK '
      003204 052111 043040 051501
      003212 020124 046103 041517
      003220 004513 000
137 003223 111 044116 041111 INHSC: .ASCIZ 'INHIBIT SLOW CLOCK '
      003230 052111 051440 047514
      003236 020127 046103 041517
      003244 004513 000
138 003247 123 041505 047524 SEC1: .ASCIZ 'SECTOR NUMBER (0=USE ALL)'
      003254 020122 052516 041115
      003262 051105 024040 036460
      003270 051525 020105 046101
      003276 024514 000

```

```

139 003302 .EVEN
140 ;
141 ;... AND THESE ARE THE DEFAULT SOFT PARAMETERS.
142 ;

```

```

143 003302 BGNSW .WORD L10003-L$SW/2
(3) 003302 000007
(3) 003304
144 003304 000000 ERRMLT: .WORD 0 ; (00) NZ = DROP AFTER N ERRORS.
145 003306 000000 PDSW: .WORD 0 ; (02) PROM DUMP SWITCH (BIT0).
146 003310 000401 PROMID: .WORD 401 ; (04) CURRENT PROM SET ID.
147 003312 000000 MPXCLK: .WORD 0 ; (06) PEAK SHIFT AND CLOCK INHIBITS.
148 003314 000000 SNGLSEC: .WORD 0 ; (10) SINGLE SECTOR TO USE (0 = ALL 6).
149 003316 000000 TLMF: .WORD 0 ; (12) G5388 TLM FLAG (1 = INSTALLED).
150 003320 160010 TCS0: .WORD 160010 ; (14) G5388 TLM CSR ADDRESS.
151 003322 ENDSW
(3) 003322

```

L10003:

```
153 .SBTTL GLOBAL EQUATES
154
155 003322 EQUALS ; GET STANDARD EQUATES.
(1) ;
(1) ; BIT DIFINITIONS
(1) ;
(1) 100000 BIT15== 100000
(1) 040000 BIT14== 40000
(1) 020000 BIT13== 20000
(1) 010000 BIT12== 10000
(1) 004000 BIT11== 4000
(1) 002000 BIT10== 2000
(1) 001000 BIT09== 1000
(1) 000400 BIT08== 400
(1) 000200 BIT07== 200
(1) 000100 BIT06== 100
(1) 000040 BIT05== 40
(1) 000020 BIT04== 20
(1) 000010 BIT03== 10
(1) 000004 BIT02== 4
(1) 000002 BIT01== 2
(1) 000001 BIT00== 1
(1) ;
(1) 001000 BIT9== BIT09
(1) 000400 BIT8== BIT08
(1) 000200 BIT7== BIT07
(1) 000100 BIT6== BIT06
(1) 000040 BIT5== BIT05
(1) 000020 BIT4== BIT04
(1) 000010 BIT3== BIT03
(1) 000004 BIT2== BIT02
(1) 000002 BIT1== BIT01
(1) 000001 BIT0== BIT00
(1) ;
(1) ; EVENT FLAG DEFINITIONS
(1) ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
(1) ;
(1) 000040 EF.START== 32. ; START COMMAND WAS ISSUED
(1) 000037 EF.RESTART== 31. ; RESTART COMMAND WAS ISSUED
(1) 000036 EF.CONTINUE== 30. ; CONTINUE COMMAND WAS ISSUED
(1) 000035 EF.NEW== 29. ; A NEW PASS HAS BEEN STARTED
(1) 000034 EF.PWR== 28. ; A POWER-FAIL/POWER-UP OCCURRED
(1) ;
(1) ;
(1) ; PRIORITY LEVEL DEFINITIONS
(1) ;
(1) 000340 PRI07== 340
(1) 000300 PRI06== 300
(1) 000240 PRI05== 240
(1) 000200 PRI04== 200
(1) 000140 PRI03== 140
(1) 000100 PRI02== 100
(1) 000040 PRI01== 40
(1) 000000 PRI00== 0
(1) ;
(1) ;OPERATOR FLAG BITS
```



```

(1)      000004      ;
(1)      000010      EVL==      4
(1)      000020      LOT==      10
(1)      000040      ADR==      20
(1)      000100      IDU==      40
(1)      000200      ISR==     100
(1)      000400      UAM==     200
(1)      001000      BOE==     400
(1)      002000      PNT==    1000
(1)      004000      PRI==    2000
(1)      010000      IXE==    4000
(1)      020000      IBE==   10000
(1)      040000      IER==   20000
(1)      100000      LOE==   40000
(1)      100000      HOE==  100000
156      ;
157      ; RL/RLV EQUATES.
158      ;
159      000004      ERRVEC= 4      ; BUS-ERROR VECTOR.
160      000000      RL11=  0      ; EQUATE THE CONTROLLER TYPES.
161      000001      RLV11=  1
162      000002      RLV12=  2      ; RLV12 WITH BAE DISABLED.
163      000003      RLV12X= 3      ; RLV12 WITH BAE ENABLED.
164
165      100000      ERR=   BIT15      ; COMPOSITE ERROR (RLCS)
166      040000      DERR=  BIT14      ; DRIVE ERROR (RLCS)
167      020000      NXM=   BIT13      ; NON-EXISTANT MEMORY (IF OPI = 0)...
168      020000      PAR=   BIT13      ; ...OR PARITY ERROR (IF OPI = 1).
169      010000      DLT=   BIT12      ; DATA LATE ERROR (IF OPI=0)...
170      010000      HNF=   BIT12      ; ...OR HEADER NOT FOUND (IF OPI=1).
171      004000      DCRC=  BIT11      ; DATA CRC ERROR (IF OPI=0)...
172      004000      HCRC=  BIT11      ; ...OR HEADER CRC ERROR (IF OPI=1).
173      002000      OPI=   BIT10      ; OPERATION INCOMPLETE (RLCS)
174      000000      DS0=    0      ; DRIVE SELECT 0 (RLCS)
175      000400      DS1=   BIT8      ; DRIVE SELECT 1 (RLCS)
176      001000      DS2=   BIT9      ; DRIVE SELECT 2 (RLCS)
177      001400      DS3=  BIT8!BIT9  ; DRIVE SELECT 3 (RLCS)
178      000200      CRDY=  BIT7      ; CONTROLLER READY (RLCS)
179      000100      INTEN= BIT6      ; INTERRUPT ENABLE (RLCS)
180      000000      NOOP=  0      ; (0) NO-OP -- RL11 ONLY.
181      000000      MAINT= 0      ; (0) MAINTENANCE -- RLV11 AND RLV12.
182      000002      WRCHK= BIT1      ; (1) WRITE CHECK FUNCTION
183      000004      GSTAT= BIT2      ; (2) GET STATUS FUNCTION
184      000006      SEEK=  BIT2!BIT1  ; (3) SEEK FUNCTION
185      000010      RDHDR= BIT3      ; (4) READ HEADER FUNCTION
186      000012      WRITE= BIT3!BIT1  ; (5) WRITE DATA FUNCTION
187      000014      READ=  BIT3!BIT2  ; (6) READ DATA FUNCTION
188      000016      RDNHDR= BIT3!BIT2!BIT1 ; (7) READ DATA WITHOUT HEADER CHECK.
189      000001      DRDY=  BIT0      ; DRIVE READY.
190
191      000010      DRST=  BIT3      ; DRIVE RESET (RLDA)
192      000002      GSBIT= BIT1      ; GET STATUS BIT (RLDA)
193      000001      MK=    BIT0      ; MARKER BIT (RLDA)
194      000004      SIGN=  BIT2      ; SIGN BIT (RLDA)
195      000100      RHHS=  BIT6      ; HEAD SELECT IN READ HEADER
196      000100      STHS=  BIT6      ; HEAD SELECT IN STATUS BACK

```

N2

CVRLB80 -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 5-2
GLOBAL EQUATES

SEQ 0026

197

000020

DAHS= BIT4

;HEAD SELECT IN SEEK

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 5-3
GLOBAL EQUATES

SEQ 0027

```
199      ;
200      ; A COUPLE OF MACROS TO REDEFINE THE ERROR CALLING CONVENTIONS
201      ; SO THAT SEQUENTIAL ERRORS ARE ASSIGNED AT ASSEMBLY TIME,
202      ; AND I DON'T HAVE TO BOTHER WITH THEM !!
203      ;
204      000000      SFN=      0      ; SYSTEM FATAL ERRORS RUN FROM 1 TO 99.
205      000143      DFN=     99.      ; DEVICE FATAL ERRORS RUN FROM 100. UP.
206
207      .MACRO SFERR ADDR,PNTR
208      SFN=SFN+1
209      ERRSF SFN,ADDR,PNTR
210      .ENDM
211
212      .MACRO DFERR ADDR,PNTR
213      DFN=DFN+1
214      ERRDF DFN,ADDR,PNTR
215      .ENDM
216      ;
217      ; ANOTHER TO START EACH TEST.
218      ;
219      .MACRO BEGIN.TEST
220      BGNTST
221      .ENDM
222      ;
223      ; AND A COUPLE OF HANDY SKIPS.
224      ;
225      000401      SKP1=     BR+1
226      000402      SKP2=     BR+2
227      000403      SKP3=     BR+3
```



```

229      .SBTTL  GLOBAL DATA
230
231 003322 000000 UNITST: .WORD 0 ; CURRENT UNIT UNDER TEST.
232 003324 000000 RLCS: .WORD 0 ; REGISTER ADDRESSES.
233 003326 000000 RLBA: .WORD 0
234 003330 000000 RLDA: .WORD 0
235 003332 000000 RLMP: .WORD 0
236 003334 000000 RLBAE: .WORD 0
237 003336 000000 BCSR: .WORD 0 ; BASE ADDRESS
238 003340 000000 BPRIOR: .WORD 0
239 003342 000000 BVEC: .WORD 0
240 003344 000000 TCSR: .WORD 0 ; G5388 TLM CSR ADDRESS.
241 003346 000000 DRIVE: .WORD 0
242 003350 000000 B.CS: .WORD 0 ; REGISTERS BEFORE FUNCTION.
243 003352 000000 B.BA: .WORD 0
244 003354 000000 B.DA: .WORD 0
245 003356 000000 B.MP: .WORD 0
246 003360 000000 B.BAE: .WORD 0
247 003362 000000 DERFLG: .WORD 0
248 003364 000000 E.CS: .WORD 0 ; REGISTERS AFTER FUNCTION.
249 003366 000000 E.BA: .WORD 0
250 003370 000000 E.DA: .WORD 0
251 003372 000000 E.MP: .WORD 0
252 003374 000000 E.MP1: .WORD 0 ; MAINT MODE CRC CHECK ONLY.
253 003376 000000 E.BAE: .WORD 0
254 003400 000000 CPUTYP: .WORD 0 ; 0 = 11/03, +NZ = 11/23, -1 = UNKNOWN.P
255 003402 000000 RLTY: .WORD 0 ; RL TYPE -- 0 = RL11, 1 = RLV11, 2(3) = RLV12.
256 003404 120001 XPOLY: .WORD 120001
257 003406 000000 BCCFBK: .WORD 0 ; LOCATION USED BY "SIMBCC"
258 003410 000000 CALBCC: .WORD 0 ; LOCATION USED BY "SIMBCC"
259 003412 000000 TEMP2: .WORD 0 ; LOCATION USED BY "SIMBCC"
260 003414 000000 TEMP3: .WORD 0 ; LOCATION USED BY "SIMBCC"
261 003416 000000 TEMP4: .WORD 0 ; LOCATION USED BY "SIMBCC"
262 003420 000000 TEMP5: .WORD 0 ; DITTO
263 003422 000000 TEMP1: .WORD 0 ; DITTO
264 003424 000000 TMP0: .WORD 0
265 003426 000000 TMP1: .WORD 0
266 003430 000000 TMP2: .WORD 0
267 003432 000000 GDDAT: .WORD 0
268 003434 000000 BDDAT: .WORD 0
269 003436 000000 INIMP: .WORD 0 ; HOLDS INITIAL MP WORD COUNT.
270 003440 000000 INIDA: .WORD 0 ; HOLDS CURRENT DA TEST WORD.
271 003442 000000 GDCRC3: .WORD 0 ; HOLDS CRC OF DA+3
272 003444 000000 GDCRC4: .WORD 0 ; HOLDA CRC OF CRC OF DA+4
273 003446 000233 OPIMN: .WORD 155.
274 003450 001212 OPIMX: .WORD 650.
275 003452 176543 HINUM: .WORD 176543
276 003454 123456 LONUM: .WORD 123456
277 003456 000000 TEMLO: .WORD 0
278 003460 000000 TEMHI: .WORD 0
279 003462 000000 DLYCNT: .WORD 0 ; 1 MSEC DELAY COUNTER.
280 003464 000000 DROPPED: .WORD 0 ; UNITS DROPPED COUNTER.
281 003466 000000 ERPOINT: .WORD 0 ; POINTS TO ONE OF THE FOLLOWING:
282 003470 000100 ERCOUNT: .BLKW 64. ; ERROR COUNTERS (ENOUGH FOR 64 UNITS).
283 003670 000000 CACREG: .WORD 0 ; IF 1, CACHE INSTALLED BUT DISABLED (11/73).
284 ;

```

```

285 ;PATTERNS USED FOR LOADING/READING REGISTERS
286 ;
287 003672 000000 BEGPAT: 000000 ;GROWING 1
288 003674 000001 000001
289 003676 000003 000003
290 003700 000007 000007
291 003702 000017 000017
292 003704 000037 000037
293 003706 000077 000077
294 003710 000177 000177
295 003712 000377 000377
296 003714 000777 000777
297 003716 001777 001777
298 003720 003777 003777
299 003722 007777 007777
300 003724 017777 017777
301 003726 037777 037777
302 003730 077777 077777
303 003732 177777 177777
304 003734 177776 177776 ;GROWING 0
305 003736 177774 177774
306 003740 177770 177770
307 003742 177760 177760
308 003744 177740 177740
309 003746 177700 177700
310 003750 177600 177600
311 003752 177400 177400
312 003754 177000 177000
313 003756 176000 176000
314 003760 174000 174000
315 003762 170000 170000
316 003764 160000 160000
317 003766 140000 140000
318 003770 100000 100000
319 003772 000000 000000
320 003774 000001 000001 ;WALKING 1
321 003776 000002 000002
322 004000 000004 000004
323 004002 000010 000010
324 004004 000020 000020
325 004006 000040 000040
326 004010 000100 000100
327 004012 000200 000200
328 004014 000400 000400
329 004016 001000 001000
330 004020 002000 002000
331 004022 004000 004000
332 004024 010000 010000
333 004026 020000 020000
334 004030 040000 040000
335 004032 100000 100000
336 004034 177777 177777 ;WALKING 0
337 004036 177776 177776
338 004040 177775 177775
339 004042 177773 177773
340 004044 177767 177767

```

341	004046	177757	177757
342	004050	177737	177737
343	004052	177677	177677
344	004054	177577	177577
345	004056	177377	177377
346	004060	176777	176777
347	004062	175777	175777
348	004064	173777	173777
349	004066	167777	167777
350	004070	157777	157777
351	004072	137777	137777
352	004074	077777	077777
353	004076	177777	177777
354	004100	000000	

ENDPAT: 000000

; PATTERNS FOR TESTING THE READ/WRITE BITS <9:1> OF THE CSR.

355			
356			
357			
358	004102	000000	
359	004104	000002	
360	004106	000004	
361	004110	000010	
362	004112	000020	
363	004114	000040	
364	004116	000100	
365	004120	000400	
366	004122	001000	
367	004124	001576	
368	004126	001574	
369	004130	001570	
370	004132	001560	
371	004134	001540	
372	004136	001500	
373	004140	001400	
374	004142	001576	
375	004144	001574	
376	004146	001566	
377	004150	001556	
378	004152	001536	
379	004154	001436	
380	004156	001136	
381	004160	000076	
382	004162	000006	
383	004164	000016	
384	004166	000036	
385	004170	000076	
386	004172	000176	
387	004174	000576	
388	004176	001576	
389	004200	000000	

; CSPAT: .WORD 0 ;SHIFTING 1

.WORD BIT1
 .WORD BIT2
 .WORD BIT3
 .WORD BIT4
 .WORD BIT5
 .WORD BIT6
 .WORD BIT8
 .WORD BIT9

;GROWING 0

;SHIFT 0

;GROWING 1

CSEND: .WORD 0

; TABLE OF DA TEST WORDS FOR MAINTENANCE MODE CRC CALCULATIONS.

390			
391			
392			
393	004202	155552	
394	004204	133330	
395	004206	066663	
396	004210	125247	

; PATCRC: 155552
 133330
 066663
 125247

397	004212	052522	052522
398	004214	177774	177774
399	004216	000374	000374
400	004220	022217	022217
401	004222	044441	044441
402	004224	166663	166663
403	004226	144441	144441
404	004230	033330	033330
405	004232	011106	011106
406	004234	070704	070704
407	004236	107065	107065
408	004240	111106	111106
409	004242	167353	167353
410	004244	156732	156732
411	004246	146311	146311
412	004250	135670	135670
413	004252	114626	114626
414	004254	104205	104205
415	004256	073564	073564
416	004260	063143	063143
417	004262	042101	042101
418	004264	031460	031460
419	004266	021037	021037
420	004270	010416	010416
421	004272	000000	000000
422			
423			
424			
425	004274	155555	155555
426	004276	133333	133333
427	004300	066666	066666
428	004302	125252	125252
429	004304	052525	052525
430	004306	177777	177777
431	004310	000000	000000
432	004312	107070	107070
433	004314	070707	070707
434	004316	144444	144444
435	004320	033333	033333
436	004322	011111	011111
437	004324	022222	022222
438	004326	044444	044444
439	004330	111111	111111
440	004332	166666	166666
441	004334	010421	010421
442	004336	021042	021042
443	004340	031463	031463
444	004342	042104	042104
445	004344	063146	063146
446	004346	073567	073567
447	004350	104210	104210
448	004352	114631	114631
449	004354	135673	135673
450	004356	146314	146314
451	004360	156735	156735
452	004362	167356	167356

CRCEND: 000000 ; 28 TEST WORDS IN THE TABLE.
; TABLE OF DATA PATTERNS FOR MAINTENANCE MODE FIFO TESTS.
; PATDAT: 155555

453 004364 000000
454
455
456
457 004366 000400
458 005366 000400
459 006366 000000

ENDDAT: 000000 ; 28 OF THESE TOO.
;
; BUFFERS FOR RLV11 MAINTENANCE FUNCTION
;
BUF1: .BLKW 256. ;MAINTENANCE MODE, DMA BUF1 => FIFO...
BUF2: .BLKW 256. ;...AND FIFO => BUF2.
BUFEND: 0

```

461      .SBTTL  INITIALIZATION CODE
462      ;
463      ; LOAD DEVICE PROTECTION NOT REQUIRED.
464      ;
465      006370      BGNPROT
(3)      006370      ; CSR OFFSET MAKE NOP          L$PROT::
466      006370      177777      .WORD  -1
467      006372      177777      .WORD  -1      ; MASS BUS OFFSET MAKE NOP
468      006374      177777      .WORD  -1      ; DRIVE OFFSET MAKE NOP
469      006376      ENDPROT
470      ;
471      ; ENTER HERE FROM DRS ON START, RESTART, CONTINUE, ETC...
472      ;
473      006376      BGNINIT
(3)      006376      ;
474      006376      READEF  #EF.PWR          ; POWER UP ??          L$INIT::
(3)      006376      012700  000034      MOV      #EF.PWR,RO
(3)      006402      104447      TRAP
475      006404      BCOMPLETE RESTART
(2)      006404      103423      BCS      RESTART
476      006406      READEF  #EF.START      ; START ??
(3)      006406      012700  000040      MOV      #EF.START,RO
(3)      006412      104447      TRAP
477      006414      BCOMPLETE START
(2)      006414      103417      BCS      START
478      006416      READEF  #EF.RESTART    ; RESTART ??
(3)      006416      012700  000037      MOV      #EF.RESTART,RO
(3)      006422      104447      TRAP
479      006424      BCOMPLETE RESTART
(2)      006424      103413      BCS      RESTART
480      006426      READEF  #EF.NEW        ; NEW PASS ??
(3)      006426      012700  000035      MOV      #EF.NEW,RO
(3)      006432      104447      TRAP
481      006434      BCOMPLETE NEWPAS
(2)      006434      103472      BCS      NEWPAS
482      006436      READEF  #EF.CONTINUE  ; CONTINUE ??
(3)      006436      012700  000036      MOV      #EF.CONTINUE,RO
(3)      006442      104447      TRAP
483      006444      BCOMPLETE 1$
(2)      006444      103401      BCS      1$
484      006446      000506      BR      NXTU          ; NONE OF THE ABOVE.
485      006450      000137  007114      JMP     CONT
486
487      006454      START:RESTART:
488      006454      000005      RESET
489      006456      005737  003306      TST     PDSW          ; PROM DUMP ??
490      006462      001402      BEQ     1$           ; BR IF NOT.
491      006464      000137  024660      JMP     ROMDUMP      ; *** PROM DUMPER, NO RETURN ***
492      006470      012737  176543  003452  1$: MOV     #176543,HINUM ;RANDOM GEN. PRIMES.
493      006476      012737  123456  003454      MOV     #123456,LONUM
494      006504      005037  003464      CLR     DROPPED      ; CLEAR UNITS DROPPED COUNTER.
495      006510      012701  003470      MOV     #ERCOUNT,R1  ; GET ERROR COUNTER POINTER.
496      006514      012700  000100      MOV     #64.,RO
497      006520      005021      2$: CLR     (R1)+      ; CLEAR ALL ERROR COUNTERS.
498      006522      005300      DEC     RO
499      006524      001375      BNE     2$

```



```

500 006526 013746 000004 ENVIRN: MOV @#4,-(SP) ; SAVE VECTOR 4...
501 006532 013746 000010 MOV @#10,-(SP) ; ...AND 10...
502 006536 010605 MOV SP,R5 ; ...AND THE STACK POINTER.
503 006540 012737 006576 000004 MOV #LSI,@#4 ; SET BUS TRAP.
504 006546 005000 CLR R0
505 006550 005737 177776 TST @#177776 ; PSW ADDRESS VALID ??
506 006554 000240 NOP ; TRAP THRU 4 IF NOT (LSI11 OR 11/2).
507 006556 012737 006572 000010 MOV #NLSI,@#10 ; SET ILLEGAL TRAP.
508 000007 MFPT= 7
509 006564 000007 MFPT ;GET PROCESSOR TYPE (IF POSSIBLE).
510 006566 000240 NOP ; TRAP THRU 10 IF NOT (NOT 11/23).
511 006570 000402 SKP2
512 006572 012700 177777 NLSI: MOV #-1,R0 ; CPU TYPE UNKNOWN.
513 006576 010037 003400 LSI: MOV R0,CPUTYP ; CPU IS LSI SOMETHING 'R OTHER.
514 006602 010506 MOV R5,SP ; RESTORE STACK...
515 006604 012637 000010 MOV (SP)+,@#10 ; ...AND VECTORS.
516 006610 012637 000004 MOV (SP)+,@#4
517 006614 004737 011306 JSR PC,.SIZE ; SIZE AVAILABLE MEMORY.
518 006620 DORPT ; REPORT OPERATING ENVIRONMENT. TRAP C$DRPT
(3) 006620 104424
519
520 006622 023737 003464 002012 NEWPAS: CMP DROPPED,L$UNIT ; UNITS STILL ALIVE ??
521 006630 002401 BLT 1$
522 006632 DOCLN ; NO, ABORT TRAP C$DCLN
(3) 006632 104444
523 006634 012737 003466 003466 1$: MOV #ERCOUNT-2,ERPOINT ;INIT THE UNIT ERROR POINTER.
524 006642 013737 003454 003456 MOV LONUM,TEMLO
525 006650 013737 003452 003460 MOV HINUM,TEMHI ; NEW PRIMES FOR NEW PASS.
526 006656 012737 177777 003322 MOV #-1,UNITST ; RESET UNIT NUMBER.
527 006664 005237 003322 NXTU: INC UNITST ; BUMP NIT NUMBER...
528 006670 062737 000002 003466 ADD #2,ERPOINT ; ...AND ERROR COUNT POINTER.
529 006676 023737 003322 002012 CMP UNITST,L$UNIT
530 006704 002346 BGE NEWPAS ;
531 006706 1$: GPHARD UNITST,R0
(3) 006706 013700 003322 MOV UNITST,R0
(3) 006712 104442 TRAP C$GPHRD
532 006714 BNCOMPLETE NXTU ; BR IF UNIT NOT THERE (DROPPED).
(2) 006714 103363 BCC NXTU
533 006716 012037 003336 2$: MOV (R0)+,BCSR ; SET UP RUN TIME P-TABLE...
534 006722 005737 036756 TST MIMIC ; ***** IF MIMIC
535 006726 001403 BEQ 3$ ; *****
536 006730 012737 037276 003336 MOV #DUMMY,BCSR ; ***** USE DUMMY REGISTERS
537 006736 012037 003342 3$: MOV (R0)+,BVEC ; ...FOR THIS UNIT.
538 006742 012037 003340 MOV (R0)+,BPRIOR
539 006746 012037 003346 MOV (R0)+,DRIVE
540 006752 012037 003402 MOV (R0)+,RLTYP ; SET CONTROLLER TYPE.
541
542 006756 013737 003316 003344 MOV TLMF,TCSR ; RUNNING WITH TLM'S ??
543 006764 001412 BEQ 5$ ; BR IF NOT.
544 006766 013737 003320 003344 MOV TCSO,TCSR ; YES, GET BASE TLM CSR...
545 006774 013700 003322 MOV UNITST,R0 ; ...AND CURRENT UNIT NUMBER.
546 007000 001404 BEQ 5$ ; BR IF UNIT IS ZERO.
547 007002 062737 000010 003344 4$: ADD #10,TCSR ; OTHERWISE, ADJUST FOR CURRENT UNIT.
548 007010 077004 SOB R0,4$
549
550 ; THE FOLLOWING CODE WAS DELETED IN ORDER TO CHANGE THE SIZING FOR ;LWL01

```

```

551 ;CPU TYPE SO THAT KDJ11-CPU TIMING LOOP COULD BE ACCOMODATED. ;LWLO1
552 ;
553 ;LWLO1 5$: TST CPUTYP
554 ;LWLO1 BGT 6$ ; BR IF 11/23.
555 ;LWLO1 BEQ 7$ ; BR IF LSI 11 OR 11/2.
556 ;LWLO1 MOV #500.,RO ; IF NEITHER, DELAY IS UNCALIBRATED.
557 ;LWLO1 BR 8$
558 ;LWLO1 6$: MOV #291.,RO ; 1MS DELAY FOR 11/23.
559 ;LWLO1 SKP2
560 ;LWLO1 7$: MOV #120.,RO ; 1MS DELAY FOR LSI11 OR 11/2.
561 ;LWLO1 8$: MOV RO,DLYCNT ; SET DELAY COUNTER.
562 007012 005000 5$: CLR RO ; START WITH CLEAR RO ;LWLO1
563 007014 005737 003400 TST CPUTYP ;LWLO1
564 007020 001427 BEQ 6$ ; BR IF LSI 11 OR 11/2 ;LWLO1
565 007022 100422 BMI 8$ ; IF MINUS, DELAY IS UNCALIBRATED ;LWLO1
566 007024 022737 000003 003400 CMP #3,CPUTYP ; IS IT AN 11/23? ;LWLO1
567 007032 001420 BEQ 7$ ; BR IF IT IS ;LWLO1
568 007034 013746 000004 MOV @#4,-(SP) ; STORE LOC 4 CONTENTS ;LWLO1
569 007040 012737 007062 000004 MOV #9,@#4 ; PREPARE FOR NON-EXISTANT TRAP ;LWLO1
570 007046 052737 001000 177746 BIS #1000,177746 ; TURN OF CACHE ;LWLO1
571 007054 012737 000001 003670 MOV #1,CACREG ; SET CACHE OFF INDICATOR ;LWLO1
572 007062 012637 000004 9$: MOV (SP)+,@#4 ; RESTORE LOC 4 ;LWLO1
573 007066 000402 BR 7$ ; IT MUST BE A KDJ11 (11/73) ;LWLO1
574 007070 062700 000321 8$: ADD #209.,RO ; TO MAKE 500. (UNCALIBRATED) ;LWLO1
575 007074 062700 000253 7$: ADD #171.,RO ; TO MAKE 291. (11/23 + 11/73) (1MSEC) ;LWLO1
576 007100 062700 000170 6$: ADD #120.,RO ; TO MAKE 120. (LSI-11 + 11/2) (1MSEC) ;LWLO1
577 007104 010037 003462 10$: MOV RO,DLYCNT ; SET DELAY COUNTER. ;LWLO1
578 007110 004737 026052 JSR PC,ADJTN ; ADJUST THE NUMBER OF TESTS TO RUN...
579 ;...ACCORDING TO CURRENT CONFIGURATION.
580 007114 013737 003456 003454 CONT: MOV TEMLO,LONUM ;RESTORE RANDOM FOR NEXT UUT
581 007122 013737 003460 003452 MOV TEMHI,HINUM ;RESTORE PRIME FOR NEXT UUT
582 007130 012737 004366 010416 MOV #BUF1,BA16 ; INIT 16 BIT BUFFER ADDRESS...
583 007136 005037 010420 CLR BA22 ;...AND 6 BIT EXTENSION (22 BITS).
584 007142 013700 003336 MOV BCSR,RO
585 007146 010037 003324 MOV RO,RLCS ; SET UNIT'S ADDRESSES.
586 007152 062700 000002 ADD #2,RO
587 007156 010037 003326 MOV RO,RLBA
588 007162 062700 000002 ADD #2,RO
589 007166 010037 003330 MOV RO,RLDA
590 007172 062700 000002 ADD #2,RO
591 007176 010037 003332 MOV RO,RLMP
592 007202 062700 000002 ADD #2,RO
593 007206 010037 003334 MOV RO,RLBAE
594 007212 013700 003342 MOV BVEC,RO
595 007216 012720 010676 MOV #RLINT,(RO)+ ; SET RL VECTOR.
596 007222 013710 003340 MOV BPRIOR,(RO)
597 007226 012737 010670 000004 MOV #TRAP4,ERRVEC ; SET TIME-OUT TRAP CATCHER.
598 007234 012737 000340 000006 MOV #PRI07,ERRVEC+2
599 007242 012737 010702 000100 MOV #CLKRTI,@#100 ; NULL THE CLOCK VECTOR.
600 007250 012737 000340 000102 MOV #PRI07,@#102
601 007256 ENDINIT
(3) 007256 L10005: TRAP C$INIT
(3) 007256 104411
602 ;
603 ; AUTO-DROP IF FLA:ADR AND UNIT DOESN'T RESPOND.
604 ;

```

```

605 007260          BGNAUTO
(3) 007260
606 007260 023737 003322 002012    CMP    UNITST,L$UNIT    ; VALID UNIT NUMBER ??
607 007266 002022                    BGE    1$                ; NO, DON'T TRY IT !!!
608 007270 005037 010672            CLR    TRPFLG           ; CLEAR TRAP FLAG.
609 007274 005777 174024            TST    @RLCS            ; TRY TO ACCESS CONTROLLER.
610 007300 000240                    240
611 007302 005737 010672            TST    TRPFLG           ; DID IT TRAP ??
612 007306 001412                    BEQ    1$                ; NO, WE'RE OK.
613 007310 013737 003324 003432    MOV    RLCS,GDDAT
614 007316          SFERR    EMO,ERR1    ; REPORT NO CONTROLLER...
(5) 007316 104454
(6) 007320 000001
(6) 007322 030710
(6) 007324 026242
615 007326          DODU    UNITST    ;...AND TELL SUPER TO DROP UNIT.
(3) 007326 013700 003322
(3) 007332 104451
616 007334          1$:    ENDAUTO
(3) 007334
(3) 007334 104461
617
618
619
620 007336          ;
(3) 007336          ; CLEAN-UP CODING SECTION (DO AT END-PASS (OR SUB PASS)).
621 007336 012700 000012            ;
622 007342 032777 000200 173754 1$:  MOV    #10.,RO          L$CLEAN::
623 007350 001004                    BIT    #CRDY,@RLCS      ; GIVE HIM ABOUT 100 MSEC...
624 007352 004537 010224            BNE    2$                ;...TO FINISH UP.
625 007356 000012                    JSR    R5,WDELAY
626 007360 077010                    10.
627 007362 012777 000200 173734 2$:  SOB    RO,1$
628 007370 023727 003402 000003    MOV    #CRDY,@RLCS      ; THEN CLEAR ALL OPTION BITS.
629 007376 001002                    CMP    RLTP,#RLV12X     ; BAE IN USE ??
630 007400 005077 173730            BNE    3$
631 007404          CLR    @RLBAE      ; IF SO, MAKE SURE IT'S CLEAR.
(3) 007404 012700 000340            3$:  SETPRI #PRI07
(3) 007410 104441
632 007412          ENDCLN
(3) 007412
(3) 007412 104412
633
634          ;
635          ; DROP UNITS -- AND KEEP TRACK OF 'EM.
636 007414          ;
(3) 007414          ;
637 007414 023737 002012 003464    CMP    L$UNIT,DROPPED   L$DU::
638 007422 001424                    BEQ    2$                ; ALL UNITS DROPPED ??
639 007424 005237 003464            INC    DROPPED           ; YUP, DO NOTHING.
640 007430          PRINTF #1$        ; NOT YET, BUMP COUNT...
(7) 007430 012746 007452                    ;...AND TELL THE MAN.
(6) 007434 012746 000001
(3) 007440 010600
(4) 007442 104417
(4) 007444 062706 000004

```



```

641 007450 000411
642 007452 040445 047125 052111 1$: BR 2$
      007460 042040 047522 050120 .ASCIZ /%AUNIT DROPPED%N/
      007466 042105 047045 000
643
644 007474 2$: .EVEN
      (3) 007474 L10010: TRAP C$DU
      (3) 007474 104453 ENDDU
645 ;
646 ; IF THEY'RE PICKED UP AGAIN, TWEAK THE COUNTER.
647 ;
648 007476 BGNAU
      (3) 007476 L$AU::
649 007476 005737 003464 TST DROPPED ; ALL UNITS BACK IN PLACE ??
650 007502 001423 BEQ 2$ ; JUST EXIT IF SO.
651 007504 005337 003464 DEC DROPPED ; NOT YET, ADJUST COUNT...
652 007510 PRINTF #1$ ;...AND TELL HIM.
      (7) 007510 012746 007532 MOV #1,-(SP)
      (6) 007514 012746 000001 MOV #1,-(SP)
      (3) 007520 010600 MOV SP,RO
      (4) 007522 104417 TRAP C$PNTF
      (4) 007524 062706 000004 ADD #4,SP
653 007530 000410
654 007532 040445 047125 052111 1$: BR 2$
      007540 040440 042104 042105 .ASCIZ /%AUNIT ADDED%N/
      007546 047045 000
655
656 007552 2$: .EVEN
      (3) 007552 L10011: TRAP C$AU
      (3) 007552 104452 ENDAU

```

```

658      .SBTTL GLOBAL SUBROUTINES
659      ;
660      ;SUBROUTINE TO CHECK FOR CONTROLLER ERRORS.
661      ;TRANSLATE THE ERROR BITS INTO PLAIN LANGUAGE AND MERGE THEM
662      ;INTO THE ERROR BUFFER (EM99) FOR SUBSEQUENT PRINTING.
663      ;DRIVE ERROR IS INCLUDED, BUT IS IGNORED IN THE DISKLESS TESTS.
664      ;
665      ;CALL: JSR    R5,GETERR
666      ;      BR     XX      ; NO ERRORS FOUND, RETURN.
667      ;      ERRXX   ; ERROR, RETURN.
668      ;
669      GETERR: MOV    R1,-(SP)      ; SAVE R1.
670      CLR    DERFLG      ;CLEAR OUT DRIVE ERROR FLAG
671      MOV    #EM99,R1    ;SET TEXT BUFFER POINTER.
672      BIT    #176000,E.CS ; ANY ERRORS THERE ??
673      BNE    1$         ; BR IF SO.
674      JSR    R5,FIX     ; NO, JUST SAY "NONE"...
675      NON
676      BR     10$       ;...AND RETURN.
677
678      1$: TST    E.CS      ;IS COMPOSITE ERROR SET ? (BETTER BE)
679      BPL    2$         ;IF NOT SOMETHING'S WRONG !!!
680      JSR    R5,FIX     ;YES, PUT "CERR" IN STRING.
681      CERR
682      2$: BIT    #DERR,E.CS ;DRIVE ERROR SET?
683      BEQ    3$         ; BR IF NOT.
684      INC    DERFLG     ;YES, SET DRV ERROR FLAG
685      JSR    R5,FIX     ; ADD "DRV" TO STRING.
686      DEMES
687      BIT    #036000,E.CS ; ANY OTHERS ???
688      BEQ    10$       ; NO, IGNORE THE DRIVE ERROR (DISKLESS).
689      TST    (R5)+      ; YES, BUMP PC TO TAKE ERROR RETURN.
690      BIT    #NXM!PAR,E.CS ; NON-EX OR PARITY ??
691      BEQ    5$         ; BR IF NOT.
692      MOV    #NXMMES,4$
693      BIT    #OPI,E.CS
694      BEQ    .+10
695      MOV    #PARGES,4$
696      JSR    R5,FIX     ; ADD "NXM" OR "PAR".
697      NXMMES
698      4$: BIT    #DLT!HNF,E.CS ; DATA LATE OR HEADER NOT FOUND ??
699      BEQ    7$         ; BR IF NOT.
700      MOV    #DLTMES,6$
701      BIT    #OPI,E.CS
702      BEQ    .+10
703      MOV    #HNFMES,6$
704      JSR    R5,FIX     ; ADD "DLT" OR "HNF".
705      DLTMES
706      6$: BIT    #DCRC!HCRC,E.CS ; EITHER CRC ??
707      BEQ    9$         ; BR IF NOT.
708      MOV    #DCRCMES,8$
709      BIT    #OPI,E.CS
710      BEQ    .+10
711      MOV    #HRCMES,8$
712      JSR    R5,FIX     ; ADD "DCRC" OR "HCRC".
713      8$: DCRCMES

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33MACY11 30(1046) 12-JUN-85 15:42 PAGE 8-1
GLOBAL SUBROUTINES

SEQ 0039

```

714 010024 032737 002000 003364 9$: BIT @OPI,E.CS ; "OPI" SET ??
715 010032 001403 BEQ 10$
716 010034 004537 010050 JSR R5,FIX ; ADD "OPI" TO STRING.
717 010040 030675 OPIMES
718 010042 105011 10$: CLRB (R1) ; TERMINATE ERROR LIST...
719 010044 012601 MOV (SP)+,R1 ; ...AND RESTORE R1.
720 010046 000205 RTS R5 ; RETURN.
721 ;
722 ;ROUTINE TO MOVE ASCII STRINGS
723 ;USES REGISTERS R1 - WHERE STRING IS BEING BUILT
724 ;
725 ; CALL JSR R5,FIX
726 ; .WORD ;ADDRESS OF STRING TO MOVE
727 ;
728 010050 012500 FIX: MOV (R5)+,R0 ;GET ADDRESS AND MOVE RETURN
729 010052 112021 1$: MOVB (R0)+,(R1)+ ;GET BYTE AND UPDATE
730 010054 001376 BNE 1$ ;WATCH 0 BYTE TERMINATOR
731 010056 105741 TSTB -(R1) ;BACK UP OVER ZERO BYTE
732 010060 000205 RTS R5 ;EXIT
733 ;
734 ;LOAD REGISTERS BEFORE FUNCTION
735 ;CALL: JSR R5,BEFORE
736 ;
737 010062 017737 173236 003350 BEFORE: MOV @RLCS,B.CS ;READ CS...
738 010070 042737 176000 003350 BIC @176000,B.CS ;...ERRORS WILL BE CLEARED ON XCT.
739 010076 017737 173224 003352 MOV @RLBA,B.BA ;READ BA
740 010104 017737 173220 003354 MOV @RLDA,B.DA ;READ DA
741 ;LWLO1 MOV INIMP,B.MP ; INITIAL MP IS THE WORD COUNT. ;LWLO1
742 010112 017737 173214 003356 MOV @RLMP,B.MP ;READ MP
743 010120 023727 003402 000003 CMP RLTP,@RLV12X
744 010126 001003 BNE 1$
745 010130 017737 173200 003360 MOV @RLBAE,B.BAE ; READ BAE IF ENABLED.
746 010136 000205 1$: RTS R5
747 ;
748 ;LOAD REGISTERS AT ERROR OR DONE.
749 ;CALL: JSR R5,AFTER
750 ;
751 010140 017737 173160 003364 AFTER: MOV @RLCS,E.CS ;READ CS
752 010146 017737 173154 003366 MOV @RLBA,E.BA ;READ BA
753 010154 017737 173150 003370 MOV @RLDA,E.DA ;READ DA
754 010162 017737 173144 003372 MOV @RLMP,E.MP ;READ MP
755 010170 005737 036756 TST MIMIC ; ***** IF MIMIC
756 010174 001003 BNE 1$ ; ***** BYPASS 2ND MP READ
757 010176 017737 173130 003374 MOV @RLMP,E.MP1 ;READ MP (MAINT 2ND CRC CHECK ONLY).
758 010204 023727 003402 000003 1$: CMP RLTP,@RLV12X
759 010212 001003 BNE 2$
760 010214 017737 173114 003376 MOV @RLBAE,E.BAE ; READ BAE IF ENABLED.
761 010222 000205 2$: RTS R5
762 ;
763 ; ROUTINE TO DELAY IN 1 MSEC INCREMENTS.
764 ; CALIBRATED FOR LSI'S ONLY.
765 ; DELAY COUNT IS 291. (443B) FOR LSI 11/23.
766 ; 120. (170B) FOR LSI 11 OR 11/2.
767 ; 500. (764B) IF CPU TYPE IS UNKNOWN.
768 ;
769 ; CALL: JSR R5,WDELAY

```



```

770      ;          N          ; DELAY N MSECS.
771      ;
772      WDELAY: MOV      R1,-(SP)
773      MOV      R2,-(SP)
774      MOV      (R5)+,R2      ; APPROX MSEC DELAY
775      2$: MOV      DLYCNT,R1  ; GET 1 MSEC TIMER.
776      3$: DEC      R1          ; START LOOP
777      BNE      3$
778      DEC      R2          ; CHECK ON MSECS REQUESTED
779      BNE      2$          ; BRANCH AND DO ANOTHER LOOP
780      4$: MOV      (SP)+,R2  ; SETUP FOR RETURN AFTER DELAY
781      MOV      (SP)+,R1
782      RTS      R5
783      ;
784      ;ROUTINE TO LOAD RLCS WITH RLV11/12 MAINTENANCE FUNCTION
785      ;EITHER FLAG DRIVEN OR INTERRUPT MODE.
786      ;
787      ;CALL: JSR      R5,LDFUN
788      ;          .WORD  MAINT      ; OR MAINT!INTEN
789      ;          .WORD  -N          ; WORD COUNT.
790      ;
791      ; LOCATIONS "BA16" AND "BA22" COMPRISE A 22 BIT PHYSICAL BUS ADDRESS
792      ; AND MUST BE SET BY THE CALLING ROUTINE PRIOR TO ENTRY.
793      ;
794      LDFUN: MOV      (R5)+,R0      ;BUILD THE FUNCTION IN R0.
795      MOV      (R5)+,INIMP      ; SAVE WORD COUNT FOR LATER.
796      BIC      #177661,R0      ;MASK GARBAGE BITS.
797      BIS      DRIVE,R0        ;INSERT DRIVE (0)...
798      BIS      #CRDY,R0        ;...AND THE READY BITS.
799      MOV      R0,@RLCS        ; LOAD UP THE CSR.
800      MOV      BA16,@RLBA      ; LOAD BUS ADDRESS <15:0>.
801      CMP      RLTP,#RLV12X    ; RLV12 WITH BAE ??
802      BEQ      1$            ; BR IF SO.
803      MOV      BA22,R0        ; NO, GET EXTENSION IF ANY...
804      BIC      #1C3,R0        ;...KEEP <17:16> ONLY...
805      ASL      R0
806      ASL      R0
807      ASL      R0
808      ASL      R0            ;...MOVE 'EM TO <5:4>...
809      BIS      R0,@RLCS        ;...AND INSERT INTO CS.
810      BR      2$
811      1$: MOV      BA22,@RLBAE  ; RLV12X, SET BA EXTENSION <21:16>.
812      2$: MOV      INIDA,@RLDA  ; LOAD DISK ADDRESS.
813      MOV      INIMP,@RLMP     ; LOAD WORD COUNT.
814      JSR      R5,BEFORE       ; SAVE STATE BEFORE FUNCTION.
815      BIC      #CRDY,@RLCS     ; XCT MAINT FUNCTION.
816      TST      MIMIC          ; ***** IF MIMIC
817      BEQ      3$            ; *****
818      JSR      PC,EMURLV      ; ***** EMULATE THE RLV FUNCTION
819      3$: RTS      R5          ;RETURN
820      ;
821      BA16: BUF1          ; 16 BIT BUFFER ADDRESS <15:0>.
822      BA22: 0            ; 6 BIT EXTENSION <21:16> IN <5:0>.
823      ;
824      ;ROUTINE TO FILL BUFFER WITH DATA PATTERNS FOR RLV MAINTENANCE.
825      ;BUF1 IS FILLED WITH 256 WORD PATTERN (OR COMPLEMENTING PATTERN).

```

```

826 ;BUF2 IS FILLED WITH 255 ZEROS AND 123456.
827 ;
828 ;CALL: JSR R5,SETPAT(SETCMP)
829 ; .WORD N ;PATTERN FOR BUFFER
830 ;
831 SETPAT: CLR R0 ; SET NO COMPLIMENT...
832 BR .+6 ;...AND SKIP NEXT.
833 SETCMP: MOV #-1,R0 ; SET COMPLIMENT.
834 MOV R1,-(SP)
835 MOV R2,-(SP)
836 MOV (R5)+,TMPO ; DATA TO STORE.
837 MOV #BUF1,R1 ;BUFFER POINTER.
838 MOV #256.,R2 ;WORD COUNT
839 1$: MOV TMPO,(R1)+
840 TST R0 ; ARE WE COMPLIMENTING ??
841 BEQ 2$ ; NO, SKIP NEXT.
842 COM TMPO ; YES, COMPLIMENT IT.
843 2$: DEC R2
844 BNE 1$
845 MOV #255.,R2 ; NOW CLEAR THE REST OF...
846 3$: CLR (R1)+ ;...THE BUFFER.
847 DEC R2
848 BNE 3$
849 MOV #123456,(R1)+ ;STORE IN LAST BUFFER WORD.
850 MOV (SP)+,R2
851 MOV (SP)+,R1
852 RTS R5

```

```

853 ;
854 ;ROUTINE TO SETUP BUFFER WITH RANDOM NUMBERS FOR RLV11 MAINT. FUNCTION
855 ;SAME PATTERN IS USED FOR EACH CONTROLLER
856 ;END OF PASS WILL CHANGE RANDOM PATTERN PRIMES
857 ; CALL JSR R5,SETRAN
858 ;

```

```

859 SETRAN: MOV R1,-(SP)
860 MOV R2,-(SP)
861 MOV #BUF1,R1 ;FIRST BUFFER START
862 MOV #256.,R2 ;BUFFER COUNT
863 1$: JSR R5,RAND ;GET RANDOM NUMBER
864 MOV HINUM,(R1)+ ;STORE IN BUFFER
865 DEC R2
866 BNE 1$
867 MOV #255.,R2 ; NOW CLEAR THE REST.
868 2$: CLR (R1)+
869 DEC R2
870 BNE 2$
871 MOV #123456,(R1)+ ;STORE IN LAST BUFFER WORD
872 MOV (SP)+,R2
873 MOV (SP)+,R1
874 RTS R5

```

```

875 ;
876 ;THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
877 ;WITH A RANGE OF 0 TO 2(+33)-1.
878 ;CALL:
879 ; JSR R5,RAND ;CALL THE ROUTINE
880 ; RETURN ;RETURN HERE THE RANDOM NUMBER
881 ; ;WILL BE IN HINUM,LONUM

```

```

882
883 010572 010146      ; RAND:  MOV R1,-(SP)      ;PUSH R1 ON STACK
884 010574 010246      MOV R2,-(SP)      ;PUSH R2 ON STACK
885 010576 010346      MOV R3,-(SP)      ;PUSH R3 ON STACK
886 010600 013703 003454  MOV LONUM,R3      ;SET R3 WITH LOW
887 010604 013701 003452  MOV HINUM,R1      ;SET R1 WITH HIGH
888 010610 012702 177771  MOV #-7,R2        ;SET SHIFT COUNTER
889 010614 006303      1$:  ASL R3           ;SHIFT R3 LEFT AND
890 010616 006101      ROL R1           ;ROTATE CARRY INTO R1 AND
891 010620 005202      INC R2           ;CHECK FOR DONE
892 010622 001374      BNE 1$          ;CONTINUE SHIFT LOOP
893 010624 063703 003454  ADD LONUM,R3      ;ADD NUMBER TO MAKE X 129
894 010630 005501      ADC R1           ;PROPOGATE CARRY
895 010632 063701 003452  ADD HINUM,R1      ;ADD NUMBER TO MAKE X 129
896 010636 062703 001057  ADD #1057,R3      ;ADD LOW CONSTANT
897 010642 005501      ADC R1           ;PROPOGATE CARRY
898 010644 062701 047401  ADD #47401,R1     ;ADD HIGH CONSTANT
899 010650 010337 003454  MOV R3,LONUM      ;SAVE R3
900 010654 010137 003452  MOV R1,HINUM      ;SAVE R1
901 010660 012603      MOV (SP)+,R3     ;POP STACK INTO R3
902 010662 012602      MOV (SP)+,R2     ;POP STACK INTO R2
903 010664 012601      MOV (SP)+,R1     ;POP STACK INTO R1
904 010666 000205      RTS R5           ;RETURN
905
906      ; ROUTINES TO SET A FLAG ON BUS-ERROR AND/OR RL INTERRUPT.
907
908 010670 005227      TRAP4: INC (PC)+
909 010672 000000      TRPFLG: 0
910 010674 000002      RTI
911
912 010676 005227      RLINT: INC (PC)+
913 010700 000000      INTFLG: 0
914 010702 000002      CLKRTI: RTI      ; USE TO DISMISS FREE-RUNNING CLOCK.
915
916      ;ROUTINE TO WAIT FOR CONTROLLER READY
917      ; THE 800 MSEC INTERVAL APPLIES TO LSI'S ONLY !!!
918      ; IN ANY CASE, IT'S JUST A KEEP-ALIVE TIMER.
919      ; ON EXIT, -4(SP) HOLDS THE REMNANTS OF THE 800. MS COUNT.
920
921 010704 012746 001440      WTCRDY: MOV #800,-(SP)      ;SET 800 MSEC TIMER.
922 010710 032777 000200 172406 1$:  BIT #CRDY,@RLCS      ;CONTROLLER READY ??
923 010716 001014      BNE 2$          ;YES, EXIT
924 010720 004537 010224      JSR R5,WDELAY    ;WAIT A WHILE
925 010724 000001      1             ;APPROX A MILLISECOND
926 010726 005316      DEC (SP)        ;CHECK IF TIME UP
927 010730 001367      BNE 1$          ;NO GO BACK
928 010732 004537 010140      JSR R5,AFTER     ; GET REGISTERS FOR ERROR.
929 010736      DFERR CRTIM,ERRO ;CONTROLLER TIMED OUT
(5) 010736 104455
(6) 010740 000144      TRAP .WORD 100
(6) 010742 030550      .WORD CRTIM
(6) 010744 026236      .WORD ERRO
930 010746 000402
931 010750 004537 010140      2$:  JSR R5,AFTER     ;GET REGISTERS
932 010754 005726      TST (SP)+       ; CLEAN THE STACK...
933 010756 000205      RTS R5           ;...AND RETURN.

```



```

934 ;
935 ;RLV11 MAINTENANCE SUBROUTINE FOR CRC CALCULATIONS
936 ;ROUTINE TO RETRIEVE PATTERN AND CALCULATE CRC OF PATTERN+3
937 ;AND CRC OF CRC OF PATTERN+4.
938 ;PATTERN IS SAVED IN "INIDA" FOR SUBSEQUENT LOADING FOR DA.
939 ;CRC OF PATTERN+3 WILL BE STORED IN "GDCRC3".
940 ;CRC OF CRC OF PATTERN+4 WILL BE STORED IN "GDCRC4".
941 ;
942 ;CALL: JSR R5,CALCRC
943 ; .WORD N ;PATTERN FOR INITIAL DA
944 ;
945 010760 012537 003440 CALCRC: MOV (R5)+,INIDA ;STORE PATTERN
946 010764 013737 003440 003422 MOV INIDA,TEMP1
947 010772 113737 003422 003420 MOVB TEMP1,TEMP5
948 011000 062737 000003 003420 ADD #3.,TEMP5 ;ADD 3 TO PATTERN
949 011006 113737 003420 003422 MOVB TEMP5,TEMP1
950 011014 013737 003422 011030 MOV TEMP1,1$
951 011022 004537 011124 JSR R5,SIMBCC ;CALCULATE EXPECTED CRC
952 011026 000020 16. ;DATA BITS
953 011030 000000 1$: .WORD 0 ;INITIAL PATTERN+3
954 011032 000000 .WORD 0
955 011034 013737 003410 003442 MOV CALBCC,GDCRC3 ;SAVE CRC OF PATTERN+3
956 011042 005237 003420 INC TEMP5 ;VALUE=PATTERN+4
957 011046 113737 003420 003422 MOVB TEMP5,TEMP1
958 011054 013737 003422 011070 MOV TEMP1,2$
959 011062 004537 011124 JSR R5,SIMBCC ;CALCULATE EXPECTED CRC
960 011066 000020 16. ;DATA BITS
961 011070 000000 2$: .WORD 0 ;INITIAL PATTERN+4
962 011072 000000 .WORD 0 ;STARTING CRC=0
963 011074 013737 003410 011110 MOV CALBCC,3$ ;STORE CRC FOR NEXT CALL
964 011102 004537 011124 JSR R5,SIMBCC ;CAL. CRC OF CRC OF DA+4
965 011106 000020 16. ;DATA BITS
966 011110 000000 3$: .WORD 0 ;CRC OF DA+4
967 011112 000000 .WORD 0 ;STARTING CRC=0
968 011114 013737 003410 003444 MOV CALBCC,GDCRC4 ;SAVE CRC OF CRC OF DA+4
969 011122 000205 RTS R5
970 ;
971 ; SUBROUTINE TO CALCULATE A CRC.
972 ;ROUTINE WILL CALCULATE A CRC-16 CRC ON A WORD OF
973 ;1-16 BITS IN LENGTH, RESULT IS RETURNED IN "CALBCC"
974 ;
975 ; CALL: JSR R5,SIMBCC
976 ; .WORD ;NUMBER OF BITS (1-16)
977 ; .WORD ;DATA FOR CRC CALCULATION
978 ; .WORD ;PREVIOUS OR STARTING CRC
979 ; (SHOULD BE ZEROED FOR START)
980 ;
981 011124 010046 SIMBCC: MOV R0,-(SP) ;SAVE R0
982 011126 010146 MOV R1,-(SP) ;SAVE R1
983 011130 010246 MOV R2,-(SP) ;SAVE R2
984 011132 012537 003412 MOV (R5)+,TEMP2 ;GET NUMBER OF BITS
985 011136 012537 003414 MOV (R5)+,TEMP3 ;GET DATA FOR CRC CALCULATION
986 011142 012537 003416 MOV (R5)+,TEMP4 ;GET STARTING CRC
987 011146 005037 003406 1$: CLR BCCFBK ;
988 011152 013700 003416 MOV TEMP4,R0 ;GET PRESENT CRC
989 011156 006037 003414 ROR TEMP3 ;ROTATE NEW DATA

```

```

990 011162 005500          ADC    R0          ;MERGE NEW WITH OLD
991 011164 032700 000001  BIT    #1,R0      ;BIT 0 SET
992 011170 001402          BEQ    2$          ;IF NOT CONTINUE
993 011172 005137 003406  COM    BCCFBK      ;
994 011176 013700 003404  2$:  MOV    XPOLY,R0   ;GET CRC POLYNOMIAL (CRC-16)
995 011202 005100          COM    R0          ;COMPLIMENT POLYNOMIAL
996 011204 040037 003406  BIC    R0,BCCFBK
997 011210 000241          CLC                    ;CLEAR CARRY
998 011212 006037 003416  ROR    TEMP4
999 011216 013700 003406  MOV    BCCFBK,R0
1000 011222 013701 003416  MOV    TEMP4,R1
1001 011226 010102          MOV    R1,R2
1002 011230 040100          BIC    R1,R0
1003 011232 043702 003406  BIC    BCCFBK,R2
1004 011236 050200          BIS    R2,R0
1005 011240 043737 003404 003416  BIC    XPOLY,TEMP4
1006 011246 050037 003416  BIS    R0,TEMP4
1007 011252 005337 003412  DEC    TEMP2
1008 011256 001333          BNE    1$
1009 011260 013737 003416 003410  MOV    TEMP4,CALBCC
1010 011266 012602          MOV    (SP)+,R2
1011 011270 012601          MOV    (SP)+,R1
1012 011272 012600          MOV    (SP)+,R0
1013 011274 000205          RTS    R5          ;RETURN

```

```

1015 .SBTTL MEMORY SIZER
1016 ;*****
1017 ;
1018 ; ROUTINE TO SIZE AVAILABLE MEMORY.
1019 ; FREELY ADAPTED FROM ".SIZE" IN SYSMAC.SML (C3).
1020 ; THIS ROUTINE MUST RESIDE WITHIN THE FIRST 24K (0-137776).
1021 ;
1022 ; USE KT IF IT'S AVAILABLE, OTHERWISE SIZE UP TO 30K IN THE
1023 ; TRADITIONAL FASHION. RETURN WITH:
1024 ;
1025 011276 000000 .MSIZE: .WORD 0 ; TOTAL MEMORY SIZE (K WORDS).
1026 011300 000000 .LSTPG: .WORD 0 ; PAGE ADDRESS (PAF) OF LAST 1K PAGE...
1027 ;...OR ZERO IF KT NOT AVAILABLE.
1028 011302 000000 .LSTAD: .WORD 0 ; LAST VIRTUAL ADDRESS IN LAST PAGE...
1029 ;...OR LAST ADDRESS UNDER 30K (IF NO KT).
1030 011304 000020 .ABUSW: .WORD 16. ; ADDRESS BUS WIDTH, 16, 18, OR 22.
1031 ;
1032 ; ALL GENERAL REGISTERS ARE USED BUT NOT SAVED.
1033 ; MEMORY PARITY ERRORS (IF ANY) ARE IGNRED.
1034 ;
1035 177572 MMR0= 177572 ; KT CONTROL REGISTERS.
1036 177574 MMR1= 177574
1037 177576 MMR2= 177576
1038 172516 MMR3= 172516
1039 172340 KIPAR0= 172340 ; KERNAL, I SPACE, PAR 0.
1040 172300 KIPDR0= KIPAR0-40 ; PDR 0.
1041 ;
1042 ;*****
1043
1044 011306 012737 000020 011304 .SIZE: MOV #16,.ABUSW ; ASSUME 16 BIT ADDRESSING.
1045 011314 013746 000004 MOV @#4,-(SP) ; SAVE BUS-ERROR...
1046 011320 013746 000006 MOV @#6,-(SP)
1047 011324 013746 000114 MOV @#114,-(SP) ;...AND PARITY VECTORS.
1048 011330 013746 000116 MOV @#116,-(SP)
1049 011334 010605 MOV SP,R5 ; SAVE STACK POINTER IN R5.
1050
1051 011336 012737 000116 000114 MOV #116,@#114 ; IGNORE PARITY ERRORS.
1052 011344 012737 000002 000116 MOV #RTI,@#116
1053 011352 012737 011376 000004 MOV #1$,@#4 ; SIZE USING "1$" IF KT ISN'T THERE.
1054 011360 012737 000340 000006 MOV #340,@#6
1055 011366 005737 177572 TST MMR0
1056 011372 000240 240
1057 011374 000417 BR .SIZKT ; OTHERWISE, SIZE USING THE KT.
1058
1059 011376 012737 011424 000004 1$: MOV #3$,@#4 ; NO KT -- SET TRAP CATCHER.
1060 011404 005002 CLR R2 ; FIRST ADDRESS (0).
1061 011406 005003 CLR R3
1062 011410 005712 2$: TST (R2) ; SIZE FROM 0 UP...
1063 011412 000240 240
1064 011414 062702 004000 ADD #4000,R2 ;...IN 1K STEPS...
1065 011420 005203 INC R3
1066 011422 000772 BR 2$ ;...UNTIL WE TRAP.
1067 011424 162702 000002 3$: SUB #2,R2 ; R2 = LAST VIRTUAL ADDRESS.
1068 011430 005001 CLR R1 ; R1 = 0 (PAF DOESN'T APPLY).
1069 011432 000513 BR .SIZXIT ; RETURN.
1070

```



```

1071 011434 012701 172340      .SIZKT: MOV      #KIPAR0,R1      ; 1ST "PAR" ADDRESS...
1072 011440 012702 172300      MOV      #KIPDR0,R2 ; ...AND IT'S "PDR"...
1073 011444 012703 000010      MOV      #8.,R3    ; ...AND THERE ARE 8 OF EACH.
1074 011450 005000              CLR      R0        ; 1ST PAGE IS ZERO.
1075 011452 010021      1$:  MOV      R0,(R1)+  ; SET PAR'S = 0, 4K, 8K ... 28K.
1076 011454 012722 077406      MOV      #77406,(R2)+ ; SET PDR'S = 4K, EX-UP, READ/WRITE.
1077 011460 062700 000200      ADD      #200,R0   ; SET NEXT PAGE PAF (+4K)...
1078 011464 077306              SOB      R3,1$     ; ...AND LOOP UNTIL ALL LOADED.
1079 011466 012741 177600      MOV      #177600,-(R1) ; PAR7 IS THE I/O PAGE (PAF).
1080 011472 005041              CLR      -(R1)    ; PAR6 WILL DO THE SIZING.
1081 011474 005003              CLR      R3       ; R3 WILL COUNT THE K'S.
1082
1083 011476 012737 011550 000004      MOV      #2$,@#4   ; TRAP TO 2$ IF NO 22 BIT SUPPORT.
1084 011504 005737 172516              TST      MMR3     ; 22 BITS SUPPORTED ??
1085 011510 012737 000020 172516      MOV      #20,MMR3 ; MUST BE, SET 22 BIT MODE...
1086 011516 012737 011564 000004      MOV      #3$,@#4   ; ...TRAP TO 3$ IF 22 BIT ADDRESS IS NXM.
1087 011524 005237 177572              INC      MMRO     ; ***** KT ON *****
1088 011530 012711 010000              MOV      #10000,(R1) ; SET PAR6 AT START OF 22 BIT LAND...
1089 011534 023737 000200 140200      CMP      @#200,140200 ; ...AND LOOK FOR WRAP-AROUND.
1090 011542 001010              BNE      3$       ; WE'RE HERE IF IT DIDN'T TRAP...
1091
1092 011544 005037 172516              CLR      MMR3     ; ...BRANCH IF IT DIDN'T WRAP-AROUND.
1093 011550 012737 000022 011304 2$:  MOV      #18...ABUSW ; WRAP-AROUND -- MUST BE 18 BITS ONLY.
1094 011556 012704 007600              MOV      #7600,R4 ; SET THE BUS WIDTH = 18.
1095 011562 000405              BR       4$       ; SET SIZER LIMIT = 124K.
1096 011564 012737 000026 011304 3$:  MOV      #22...ABUSW ; WE HAVE A REAL 22 BIT ADDRESS SPACE.
1097 011572 012704 170000              MOV      #170000,R4 ; SET SIZER LIMIT = 1920K.
1098 011576
1099 011576 012737 011624 000004 4$:  .SIZMEM: MOV     #2$,@#4   ; TRAP TO 2$ WHEN DONE SIZING.
1100 011604 005011              CLR      (R1)    ; SET PAR6 AT 1ST PAGE (0).
1101 011606 005737 140000      1$:  TST      140000  ; SIZE USING PAR6 (+0)...
1102 011612 062711 000040              ADD      #40,(R1) ; ...IN 1K STEPS.
1103 011616 005203              INC      R3
1104 011620 021104              CMP      (R1),R4 ; REACHED LIMIT ??
1105 011622 103771              BLO     1$       ; LOOP IF NOT.
1106 011624 011100      2$:  MOV      (R1),R0 ; DONE, SAVE FINAL PAR6...
1107 011626 012711 001400              MOV      #1400,(R1) ; ...AND RESET IT TO BANK 6 (24K).
1108 011632 005037 177572              CLR      MMRO     ; ***** KT OFF *****
1109 011636 012737 011650 000004      MOV      #3$,@#4   ; THERE MAY NOT BE AN MMR3 ;LWLO1
1110 011644 005037 172516              CLR      MMR3     ; 22 BIT ADDRESSING OFF ;LWLO1
1111 011650 010001      3$:  MOV      R0,R1   ; RECOVER SIZING RESULT.
1112 011652 162701 000040              SUB      #40,R1   ; R1 = LAST 1K PAGE (PAF).
1113 011656 012702 003776              MOV      #3776,R2 ; R2 = LAST ADDR IN THAT PAGE.
1114
1115 011662 010137 011300      .SIZXIT: MOV     R1,.LSTPG ; RETURN LAST PAGE (PAF)...
1116 011666 010237 011302              MOV      R2,.LSTAD ; ...LAST VIRTUAL ADDRESS...
1117 011672 010337 011276              MOV      R3,.MSIZE ; ...AND TOTAL MEMORY SIZE (K).
1118 011676 010506              MOV      R5,SP    ; RECOVER OUR STACK POINTER...
1119 011700 012637 000116              MOV      (SP)+,@#116 ; ...AND THE ERROR VECTORS.
1120 011704 012637 000114              MOV      (SP)+,@#114
1121 011710 012637 000006              MOV      (SP)+,@#6
1122 011714 012637 000004              MOV      (SP)+,@#4
1123 011720 000207              RTS      PC       ; ...AND RETURN.

```

```

1125      .SBTTL  REPORT ENVIRONMENT
1126      ;
1127      ; SUBROUTINE TO DISPLAY THE CURRENT OPERATING ENVIRONMENT.
1128      ; CALLED ON EVERY START/RESTART, OR DIRECTLY VIA "PRI".
1129      ;
1130      BGNRPT
1131      (3) 011722
1132      011722 105037 012162 1$: CLRB  MSGNOT
1133      011726 005737 011300      TST   .LSTPG
1134      011732 001003          BNE   2$
1135      011734 112737 000040 012162 2$: MOVB  #40,MSGNOT
1136      011742          PRINTF #MSG2,#MSGNOT ; MMU
1137      (8) 011742 012746 012162          MOV  #MSGNOT,-(SP)
1138      (7) 011746 012746 012042          MOV  #MSG2,-(SP)
1139      (6) 011752 012746 000002          MOV  #2,-(SP)
1140      (3) 011756 010600          MOV  SP,RO
1141      (4) 011760 104417          TRAP C$PNTF
1142      (4) 011762 062706 000006          ADD  #6,SP
1143      PRINTF #MSG3,.MSIZE ; MEM SIZE.
1144      (8) 011766 013746 011276          MOV  .MSIZE,-(SP)
1145      (7) 011772 012746 012072          MOV  #MSG3,-(SP)
1146      (6) 011776 012746 000002          MOV  #2,-(SP)
1147      (3) 012002 010600          MOV  SP,RO
1148      (4) 012004 104417          TRAP C$PNTF
1149      (4) 012006 062706 000006          ADD  #6,SP
1150      3$: PRINTF #MSG4,.ABUSW ; ADDRESS SPACE.
1151      (8) 012012 013746 011304          MOV  .ABUSW,-(SP)
1152      (7) 012016 012746 012125          MOV  #MSG4,-(SP)
1153      (6) 012022 012746 000002          MOV  #2,-(SP)
1154      (3) 012026 010600          MOV  SP,RO
1155      (4) 012030 104417          TRAP C$PNTF
1156      (4) 012032 062706 000006          ADD  #6,SP
1157      1138 012036          EXIT  RPT
1158      (4) 012036 000167          .WORD J$JMP
1159      (3) 012040 000126          .WORD L10012-2-.
1160      1139 012042 047045 040445 020040 MSG2: .ASCIZ '%N%A MMU%T%A AVAILABLE'
1161      012050 046515 022525 022524
1162      012056 020101 053101 044501
1163      012064 040514 046102 000105
1164      1140 012072 047045 040445 020040 MSG3: .ASCIZ '%N%A MEMORY SIZE %D4%A KW'
1165      012100 042515 047515 054522
1166      012106 051440 055111 020105
1167      012114 042045 022464 020101
1168      012122 053513 000
1169      1141 012125 045 022516 020101 MSG4: .ASCIZ '%N%A %D2%A BIT ADDRESSING%N'
1170      012132 022440 031104 040445
1171      012140 041040 052111 040440
1172      012146 042104 042522 051523
1173      012154 047111 022507 000116
1174      1142 012162 047040 052117 000 MSGNOT: .ASCIZ ' NOT'
1175      1143 012170          .EVEN
1176      1144 012170          ENDRPT
1177      (3) 012170          L10012: TRAP  C$RPT
1178      (3) 012170 104425

```

```

1146 .SBTTL
1147 .SBTTL RL DISKLESS CONTROLLER TESTS.
1148 .SBTTL
1149 .SBTTL 1 -- RLCS ADDRESSABILITY.
1150
1151 012172 STARS
(2) ;*****
1152 ;TEST TO SEE IF WE CAN ADDRESS THE CONTROL
1153 ;AND STATUS REGISTER. IF WE TRAP WE WILL REPORT
1154 ;THE ERROR AND ABORT. AFTER THIS TEST WE ONLY KNOW
1155 ;THAT WE CAN ADDRESS THE REGISTER.
1156 012172 STARS
(2) ;*****
1157 012172 BEGIN.TEST
(4) 012172 T1::
1158 012172 005037 010672 1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
1159 012176 012777 177777 171120 MOV #177777,@RLCS ; WRITE RLCS.
1160 012204 000240 000240 240,240
1161 012210 005777 171110 TST @RLCS ; READ RLCS.
1162 012214 000240 000240 240,240
1163 012220 005737 010672 TST TRPFLG ; EITHER ONE TRAP ??
1164 012224 001413 BEQ 3$ ; NO, PROCEED.
1165 012226 013737 003324 003432 MOV RLCS,GDDAT ;SET UP ERROR DATA
1166 012234 SFERR EM1,ERR1 ;BUS TIMEOUT IN ADDRESSING RLCS
(5) 012234 104454 TRAP C$ERSF
(6) 012236 000002 .WORD 2
(6) 012240 030760 .WORD EM1
(6) 012242 026242 .WORD ERR1
1167 012244 DODU UNITST ; DROP...
(3) 012244 013700 003322 MOV UNITST,R0
(3) 012250 104451 TRAP C$DODU
1168 012252 DOCLN ;...AND ABORT.
(3) 012252 104444 TRAP C$DCLN
1169 012254 3$: ENDTST L10013:
(3) 012254 104401 TRAP C$ETST

```



```

1171 .SBTTL 2 -- RLBA ADDRESSABILITY.
1172
1173 012256 STARS
(2) ;*****
1174 ;TEST TO SEE IF WE CAN ADDRESS THE BUS ADDRESS
1175 ;REGISTER. IF WE TRAP WE WILL REPORT THE ERROR
1176 ;AND ABORT. AFTER THIS TEST WE ONLY KNOW THAT
1177 ;WE CAN ADDRESS THE REGISTER.
1178 012256 STARS
(2) ;*****
1179 012256 BEGIN.TEST
(4) 012256 T2::
1180 012256 005037 010672 1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
1181 012262 012777 177777 171036 MOV #177777,@RLBA ; WRITE RLBA.
1182 012270 000240 000240 240,240
1183 012274 005777 171026 TST @RLBA ; READ RLBA.
1184 012300 000240 000240 240,240
1185 012304 005737 010672 TST TRPFLG ; ANY TRAPS ??
1186 012310 001413 BEQ 3$ ;NO, CONTINUE
1187 012312 013737 003326 003432 MOV RLBA,GDDAT ;SETUP ERROR DATA
1188 012320 SFERR EM2,ERR1 ;BUS TIMEOUT IN ADDRESSING RLBA
(5) 012320 104454 TRAP C$ERSF
(6) 012322 000003 .WORD 3
(6) 012324 031005 .WORD EM2
(6) 012326 026242 .WORD ERR1
1189 012330 DODU UNITST ; DROP...
(3) 012330 013700 003322 MOV UNITST,RO
(3) 012334 104451 TRAP C$DODU
1190 012336 DOCLN ;...AND ABORT.
(3) 012336 104444 TRAP C$DCLN
1191 012340 3$: ENDTST
(3) 012340 L10014:
(3) 012340 104401 TRAP C$ETST

```

.SBTTL 3 -- RLDA ADDRESSABILITY.

STARS

;TEST TO SEE IF WE CAN ADDRESS THE DISK ADDRESS
;REGISTER IF WE TRAP WE WILL REPORT THE ERROR
;AND ABORT. AFTER THIS TEST WE ONLY KNOW THAT
;WE CAN ADDRESS THE REGISTER.
STARS

STARS

BEGIN.TEST

T3::

1193
1194
1195 012342
(2)
1196
1197
1198
1199
1200 012342
(2)
1201 012342
(4) 012342
1202 012342 005037 010672
1203 012346 012777 177777 170754
1204 012354 000240 000240
1205 012360 005777 170744
1206 012364 000240 000240
1207 012370 005737 010672
1208 012374 001413
1209 012376 013737 003330 003432
1210 012404
(5) 012404 104454
(6) 012406 000004
(6) 012410 031032
(6) 012412 026242
1211 012414
(3) 012414 013700 003322
(3) 012420 104451
1212 012422
(3) 012422 104444
1213 012424
(3) 012424
(3) 012424 104401

1\$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
MOV #177777,@RLDA ; WRITE RLDA
240,240
TST @RLDA ; READ RLDA
240,240
TST TRPFLG ; ANY TRAPS ??
BEQ 3\$;NO, CONTINUE
MOV RLDA,GDDAT ;SETUP ERROR INFO
SFERR EM3,ERR1 ;BUS TIMEOUT IN ADDRESSING RLDA

DODU UNITST ; DROP...

DOCLN ;...AND ABORT.

TRAP C\$ERSF
.WORD 4
.WORD EM3
.WORD ERR1

MOV UNITST,RO
TRAP C\$DODU

TRAP C\$DCLN

3\$: ENDTST

L10015:

TRAP C\$ETST

1215
1216
1217 012426
(2)
1218
1219
1220
1221
1222 012426
(2)
1223 012426
(4) 012426
1224 012426 005037 010672
1225 012432 012777 177777 170672
1226 012440 000240 000240
1227 012444 005777 170662
1228 012450 000240 000240
1229 012454 005737 010672
1230 012460 001413
1231 012462 013737 003332 003432
1232 012470
(5) 012470 104454
(6) 012472 000005
(6) 012474 031057
(6) 012476 026242
1233 012500
(3) 012500 013700 003322
(3) 012504 104451
1234 012506
(3) 012506 104444
1235 012510
(3) 012510
(3) 012510 104401

.SBTTL 4 -- RLMP ADDRESSABILITY.

STARS

;;*****
;TEST TO SEE IF WE CAN ADDRESS THE MULTIPURPOSE
;REGISTER. IF WE TRAP WE WILL REPORT THE ERROR AND
;ABORT. AFTER THIS TEST WE ONLY KNOW THAT WE CAN
;ADDRESS THE REGISTER.

STARS

;;*****

BEGIN.TEST

1\$: CLR TRPFLG ;CLEAR TRAP OCCURANCE T4:;
MOV #177777,@RLMP ;WRITE RLMP
240,240
TST @RLMP ; READ RLMP
240,240
TST TRPFLG ; ANY TRAPS ??
BEQ 3\$;NO, CONTINUE
MOV RLMP,GDDAT ;SET UP ERROR INFO
SFERR EM4,ERR1 ;BUS TIMEOUT IN ADDRESSING RLMP

TRAP C\$ERSF
.WORD 5
.WORD EM4
.WORD ERR1

DODU UNITST ; DROP...

MOV UNITST,RO
TRAP C\$DODU

DOCLN ;...AND ABORT.

TRAP C\$DCLN

3\$: ENDTST

L10C16: TRAP C\$ETST

.SBTTL 5 -- RLBAE ADDRESSABILITY (RLV12 ONLY).

1237
1238
1239 012512
(2)
1240
1241
1242 012512
(2)
1243 012512
(4) 012512
1244 012512 023727 003402 000002
1245 012520 103444
1246 012522 005037 010672
1247 012526 013737 003334 003432
1248 012534 012777 177777 170572
1249 012542 000240 000240
1250 012546 005777 170562
1251 012552 000240 000240
1252 012556 023727 003402 000003
1253 012564 001013
1254
1255 012566 005737 010672
1256 012572 001417
1257 012574
(5) 012574 104454
(6) 012576 000006
(6) 012600 031104
(6) 012602 026242
1258 012604
(3) 012604 013700 003322
(3) 012610 104451
1259 012612
(3) 012612 104444
1260
1261 012614 005737 010672
1262 012620 001004
1263 012622
(5) 012622 104455
(6) 012624 000145
(6) 012626 031132
(6) 012630 026242
1264 012632
(3) 012632
(3) 012632 104401

STARS
;*****
; THE DEFAULT (SHIP) CONFIGURATION REQUIRES THAT THE
; BAE REGISTER BE ENABLED. CHECK IT AND REPORT ACCORDINGLY.
STARS
;*****
BEGIN.TEST
T5::
CMP RLTYP,#RLV12 ; RLV12 ??
BLO 4\$; NO, SKIP THIS TEST.
1\$: CLR TRPFLG ; CLEAR TRAP OCCURANCE
MOV RLBAE,GDDAT ; SET ADDRESS IN CASE OF ERROR.
MOV #-1,@RLBAE ; WRITE BAE.
240,240
TST @RLBAE ; READ BAE.
240,240
CMP RLTYP,#RLV12X ; DEFAULT CONFIG (W/BAE) ??
BNE 3\$; BR IF NOT.
2\$: TST TRPFLG ; BAE SHOULD HAVE ANSWERED -- DID IT ??
BEQ 4\$; BR IF SO.
SFERR EM4A,ERR1 ;BUS TIMEOUT IN ADDRESSING RLBAE
DODU UNITST ; DROP...
DOCLN ;...AND ABORT.
3\$: TST TRPFLG ; BAE SHOULD HAVE TRAPPED -- DID IT ??
BNE 4\$; BR IF SO.
DFERR EM4B,ERR1 ; BAE IS NOT DISABLED.
4\$: ENDTST
L10017:
TRAP C\$ERSF
.WORD 6
.WORD EM4A
.WORD ERR1
MOV UNITST,RO
TRAP C\$DODU
TRAP C\$DCLN
TRAP C\$ERDF
.WORD 101
.WORD EM4B
.WORD ERR1
TRAP C\$ETST

1266
1267
1268 012634
(2)
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278 012634
(2)
1279 012634
(4) 012634
1280 012634
(3) 012634 012700 000340
(3) 012640 104441
1281 012642 012777 000377 170454
1282 012650 012737 000200 003432
1283 012656 032777 040000 170440
1284 012664 001403
1285 012666 052737 140000 003432
1286 012674 012777 177777 170424 1\$:
1287 012702 012777 177777 170420
1288 012710 023727 003402 000003
1289 012716 001003
1290 012720 012777 177777 170406
1291 012726 004537 010224 2\$:
1292 012732 000372
1293 012734 000005
1294 012736 004537 010224 3\$:
1295 012742 000372
1296 012744 017737 170354 003434
1297 012752 042737 000001 003434
1298 012760 023737 003434 003432
1299 012766 001404
1300 012770
(5) 012770 104455
(6) 012772 000146
(6) 012774 033363
(6) 012776 026272
1301 013000 005037 003432 4\$:
1302 013004 017737 170316 003434
1303 013012 001404
1304 013014
(5) 013014 104455
(6) 013016 000147
(6) 013020 033417
(6) 013022 026272
1305 013024 017737 170300 003434 5\$:
1306 013032 001404
1307 013034
(5) 013034 104455

.SBTTL 6 -- BUS RESET OF ALL REGISTERS.

STARS

;TEST THAT A BUS RESET WILL CLEAR THE PROPER BITS.
;IN THE CONTROL AND STATUS REGISTER, THOSE BITS ARE
;1-6,8,9,10,11,12,13,15. BIT 15 WILL CLEAR ONLY
;IF BIT 14 (DRIVE ERROR IS NOT SET). BIT 0 (DRIVE READY)
;IS A DON'T CARE. IF AT THE START UP THIS TEST BIT
;14 (DRIVE ERROR) IS SET WE WILL INSIST IF IS THERE AFTER
;THE "RESET" ALONG WITH BIT 15 (COMPOSITE ERROR). BITS
;15-10 ARE NOT WRITEABLE.
; THE OTHER REGISTERS SHOULD GO TO ZERO.

STARS

BEGIN.TEST

T6::

SETPRI #PRI07 ;PRIORITY TO SEVEN
MOV #377,@RLCS ;LOAD ALL RLCS LOADABLE BITS
MOV #CRDY,GDDAT ;SETUP EXPECTED CSR.
BIT #DERR,@RLCS ;DRIVE ERR SET?
BEQ 1\$;IF NOT DON'T EXPECT IT
BIS #DERR!ERR,GDDAT ;IT'S SET, INIT BETTER NOT CLR
MOV #-1,@RLBA ; LOAD ALL BITS IN THE OTHERS.
MOV #-1,@RLDA
CMP RLTP,@RLV12X ; RLV12X ??
BNE 2\$; NO, SKIP NEXT.
MOV #-1,@RLBAE
JSR R5,WDELAY ; DELAY BEFORE...
250.
RESET
JSR R5,WDELAY ;...AND AFTER.
250.
MOV @RLCS,BDDAT ;READ RLCS
BIC #DRDY,BDDAT ;CLEAR OUT DRDY - DON'T CARE
CMP BDDAT,GDDAT ;DID INIT WORK ON CSR.
BEQ 4\$;YES, BRANCH
DFERR EM69,ERR2 ;BUS-INIT FAILED ON CSR.
TRAP C\$ERDF
.WORD 102
.WORD EM69
.WORD ERR2
CLR GDDAT ; EXPECT 0 IN THE OTHERS.
MOV @RLBA,BDDAT
BEQ 5\$
DFERR EM70,ERR2 ; BUS-INIT FAILED ON RLBA.
TRAP C\$ERDF
.WORD 103
.WORD EM70
.WORD ERR2
MOV @RLDA,BDDAT
BEQ 6\$
DFERR EM71,ERR2 ; BUS-INIT FAILED ON RLDA.
TRAP C\$ERDF

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 14-1
6 -- BUS RESET OF ALL REGISTERS.

SEQ 0054

(6)	013036	000150									.WORD	104
(6)	013040	033453									.WORD	EM71
(6)	013042	026272									.WORD	ERR2
1308	013044	023727	003402	000003	6\$:	CMP	RLTYP,#RLV12X		; RLV12X ??			
1309	013052	001010				BNE	7\$; WE'RE DONE IF NOT.			
1310	013054	017737	170254	003434		MOV	@RLBAE,BDDAT					
1311	013062	001404				BEQ	7\$					
1312	013064					DFERR	EM71A,ERR2		; BUS-INIT FAILED ON RLBAE.			
(5)	013064	104455								TRAP	C\$ERDF	
(6)	013066	000151								.WORD	105	
(6)	013070	033507								.WORD	EM71A	
(6)	013072	026272								.WORD	ERR2	
1313	013074				7\$:	ENDTST						
(3)	013074									L10020:		
(3)	013074	104401								TRAP	C\$ETST	

1315
1316
1317 013076
(2)
1318
1319
1320
1321
1322 013076
(2)
1323 013076
(4) 013076
1324 013076 012703 004102
1325 013102
(3) 013102 104404
1326 013104 011337 003432
1327 013110 052737 000200 003432
1328 013116 013777 003432 170200
1329 013124 032777 040000 170172
1330 013132 001403
1331 013134 052737 140000 003432
1332 013142 017737 170156 003434
1333 013150 042737 000001 003434
1334 013156 023737 003432 003434
1335 013164 001404
1336 013166
(5) 013166 104455
(6) 013170 000152
(6) 013172 031201
(6) 013174 026272
1337 013176
(3) 013176 104410
(3) 013200 000012
1338 013202 005723
1339 013204 020327 004200
1340 013210 001335
1341 013212
(3) 013212
(3) 013212 104405
1342 013214
(3) 013214
(3) 013214 104401

.SBTTL 7 -- READ WRITE OF RLCS.

STARS

```

;*****
;TEST THAT WE CAN WRITE/READ BITS 8,9 AND BITS 6-1
;OF THE CONTROL AND STATUS REGISTER. BITS 15-10 AND 0
;ARE DON'T CARE BITS AT THIS TIME AND BIT 7
;(CONTROLLER READY) IS ALWAYS WRITTEN TO A ONE.
STARS
;*****

```

BEGIN.TEST

```

;*****
;SET UP TABLE POINTER OF PATTERNS
;*****START OF SEGMENT*****
T7::
TRAP C$BSEG
MOV #CSPAT,R3 ;GET PATTERN INTO GDDAT
BGNSEG ;INSURE GO IS SET
1$: MOV (R3),GDDAT ;LOAD RLCS (CONTROL AND STATUS)
BIS #200,GDDAT ;IF DRIVE ERROR PRESENT
MOV GDDAT,@RLCS ;THEN EXPECT DRIVE AND
BIT #DERR,@RLCS ;COMPOSITE ERROR
BEQ 2$ ;READ RLCS BACK
BIS #ERR!DERR,GDDAT ;IGNORE DRIVE READY
2$: MOV @RLCS,BDDAT ;DID WE READ WHAT WE LOADED
BIC #DRDY,BDDAT ;YES, THEN BRANCH
CMP GDDAT,BDDAT ;WRONG DATA IN RLCS
BEQ 3$
DFERR EM5,ERR2
TRAP C$ERDF
.WORD 106
.WORD EM5
.WORD ERR2
3$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA:LOE).
TRAP C$ESCAPE
.WORD 10000$-.
TST (R3)+ ;BUMP FOR NEXT PATTERN
CMP R3,#CSEND ;CHECK FOR END
BNE 1$ ;NOT END, LOAD NEXT PATTERN
ENDSEG ;*****END OF SEGMENT*****
10000$: TRAP C$ESEG
ENDTST L10021: TRAP C$ETST

```

```

1344 .SBTTL 8 -- READ WRITE OF RLBA.
1345
1346 013216 STARS
(2) ;*****
1347 ;TEST THAT WE CAN WRITE/READ BITS 15 THRU 1 OF THE
1348 ;BUS ADDRESS REGISTER. FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
1349 ;GROWING 0 AND SHIFTING 0. BIT 0 IS ALSO LOADED BUT
1350 ;SHOULD ALWAYS COME BACK AS 0
1351 013216 STARS
(2) ;*****
1352 013216 BEGIN.TEST
(4) 013216 T8::
1353 013216 012703 003672 MOV #BEGPAT,R3 ;GET START OF PATTERN LIST
1354 013222 BGNSEG ;****START OF SEGMENT**** TRAP C$BSEG
(3) 013222 104404
1355 013224 011337 003432 1$: MOV (R3),GDDAT ;GET PATTERN TO SEND
1356 013230 023727 003402 000000 CMP RL11,#RL11 ; RL11 ??
1357 013236 001003 BNE 2$ ;NO.
1358 013240 042737 000001 003432 BIC #BIT0,GDDAT ;YES, KEEP RLBA EVEN (UNIBUS)
1359 013246 013777 003432 170052 2$: MOV GDDAT,@RLBA ;LOAD PATTERN TO BUS ADDRESS
1360 013254 017737 170046 003434 MOV @RLBA,BDDAT ;READ IT BACK
1361 013262 023737 003432 003434 CMP GDDAT,BDDAT ;IS IT CORRECT?
1362 013270 001404 BEQ 3$ ;IF SO, BRANCH
1363 013272 DFERR EM6,ERR2 ;DATA WRONG IN RLBA
(5) 013272 104455 TRAP C$ERDF
(6) 013274 000153 .WORD 107
(6) 013276 031252 .WORD EM6
(6) 013300 026272 .WORD ERR2
1364 013302 3$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE
(3) 013302 104410 .WORD 10000$-.
(3) 013304 000012
1365 013306 005723 TST (R3)+ ;BUMP FOR NEXT PATTERN
1366 013310 020327 004100 CMP R3,#ENDPAT ;CHECK FOR END
1367 013314 001343 BNE 1$ ; LOOP TIL DONE.
1368 013316 ENDSEG ;****END OF SEGMENT****
(3) 013316 10000$: TRAP C$ESEG
(3) 013316 104405
1369 013320 ENDTST
(3) 013320 L10022: TRAP C$ETST
(3) 013320 104401

```

1371
1372
1373 013322
(2)
1374
1375
1376
1377 013322
(2)
1378 013322
(4) 013322
1379 013322 012703 003672
1380 013326
(3) 013326 104404
1381 013330 011337 003432
1382 013334 013777 003432 167766
1383 013342 017737 167762 003434
1384 013350 023737 003432 003434
1385 013356 001404
1386 013360
(5) 013360 104455
(6) 013362 000154
(6) 013364 031300
(6) 013366 026272
1387 013370
(3) 013370 104410
(3) 013372 000012
1388 013374 005723
1389 013376 020327 004100
1390 013402 001352
1391 013404
(3) 013404
(3) 013404 104405
1392 013406
(3) 013406
(3) 013406 104401

.SBTTL 9 -- READ WRITE OF RLDA.

STARS

```

;*****
;TEST THAT WE CAN WRITE/READ THE DISK ADDRESS REGISTER
;ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
STARS

```

;*****

BEGIN.TEST

```

MOV #BEGPAT,R3 ;SET UP POINTER TO PATTERN LIST
BGNSEG ;****START OF SEGMENT****

```

T9::

```

1$: MOV (R3),GDDAT ;GET PATTERN
MOV GDDAT,@RLDA ;LOAD PATTERN IN DA
MOV @RLDA,BDDAT ;READ PATTERN BACK
CMP GDDAT,BDDAT ;IS IT CORRECT?
BEQ 2$ ;BRANCH IF CORRECT
DFERR EM7,ERR2 ;WRONG DATA IN RLDA

```

TRAP C\$BSEG

TRAP C\$ERDF
.WORD 108
.WORD EM7
.WORD ERR2

```

2$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA:LOE).

```

TRAP C\$ESCAPE
.WORD 10000\$-

```

TST (R3)+ ;BUMP POINTER
CMP R3,#ENDPAT ;AT END OF PATTERNS?
BNE 1$ ;NO, BRANCH BACK
ENDSEG ;****END OF SEGMENT****

```

10000\$:

TRAP C\$ESEG

ENDTST

L10023:

TRAP C\$ETST

.SBTTL 10 -- READ WRITE OF RLBAE (RLV12 ONLY).

1394
1395
1396 013410

(2)
1397
1398
1399
1400

1401 013410

(2)
1402 013410

(4) 013410

1403 013410 023727 003402 000003

1404 013416 001035

1405 013420 012703 003672

1406 013424

(3) 013424 104404

1407 013426 011337 003432

1408 013432 013777 003432 167674

1409 013440 017737 167670 003434

1410 013446 042737 177700 003432

1411 013454 023737 003432 003434

1412 013462 001404

1413 013464

(5) 013464 104455

(6) 013466 000155

(6) 013470 031326

(6) 013472 026272

1414 013474

(3) 013474 104410

(3) 013476 000012

1415 013500 005723

1416 013502 020327 004100

1417 013506 001347

1418 013510

(3) 013510

(3) 013510 104405

1419 013512

(3) 013512

(3) 013512 104401

STARS

;TEST THAT WE CAN WRITE/READ THE BUS ADDRESS EXTENSION REGISTER.
;ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
; ONLY THE LOW 6 BITS ARE TESTED RLBAE<5:0>

STARS

BEGIN.TEST

T10::

CMP RLTP,#RLV12X ; BAE ENABLED ??
BNE 3\$; NO, DON'T EVEN TRY.
MOV #BEGPAT,R3 ;SET UP POINTER TO PATTERN LIST
BGNSEG ;****START OF SEGMENT****

TRAP C\$BSEG

1\$:

MOV (R3),GDDAT ;GET PATTERN
MOV GDDAT,@RLBAE ;LOAD PATTERN IN BAE
MOV @RLBAE,BDDAT ;READ PATTERN BACK
BIC #C77,GDDAT ; ** ONLY 5:0 ARE VALID BITS.
CMP GDDAT,BDDAT ;IS IT CORRECT?
BEQ 2\$;BRANCH IF CORRECT
DFERR EM8,ERR2 ;WRONG DATA IN RLBAE

TRAP C\$ERDF
.WORD 109
.WORD EM8
.WORD ERR2

2\$:

ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE).

TRAP C\$ESCAPE
.WORD 10000\$-

TST (R3)+ ;BUMP POINTER
CMP R3,#ENDPAT ;AT END OF PATTERNS?
BNE 1\$;NO, BRANCH BACK
ENDSEG ;****END OF SEGMENT****

10000\$:

TRAP C\$ESEG

3\$:

ENDTST

L10024:

TRAP C\$ETST

.SBTTL 11 -- BIS AND BIC OF RLCS.

1421
1422
1423 013514
(2)
1424
1425
1426
1427
1428 013514
(2)
1429 013514
(4) 013514
1430 013514 012703 004102
1431 013520
(3) 013520 104404
1432 013522 012777 000200 167574 1\$:
1433 013530 011337 003432
1434 013534 052737 000200 003432
1435 013542 051377 167556
1436 013546 032777 040000 167550
1437 013554 001403
1438 013556 052737 140000 003432
1439 013564 017737 167534 003434 2\$:
1440 013572 042737 000001 003434
1441 013600 023737 003434 003432
1442 013606 001404
1443 013610
(5) 013610 104455
(6) 013612 000156
(6) 013614 032671
(6) 013616 026272
1444 013620 3\$:
(3) 013620 104410
(3) 013622 000114
1445 013624 012777 001776 167472 4\$:
1446 013632 012737 001776 003432
1447 013640 041337 003432
1448 013644 041377 167454
1449 013650 032777 040000 167446
1450 013656 001403
1451 013660 052737 140000 003432
1452 013666 017737 167432 003434 5\$:
1453 013674 042737 000001 003434
1454 013702 023737 003434 003432
1455 013710 001404
1456 013712
(5) 013712 104455
(6) 013714 000157
(6) 013716 032737
(6) 013720 026272
1457 013722 6\$:
(3) 013722 104410
(3) 013724 000012
1458 013726 005723
1459 013730 020327 004200
1460 013734 001272

STARS

```

;*****
;TEST THAT WE CAN USE THE "BIS" AND "BIC" INSTRUCTIONS ON THE CONTROL
;AND STATUS REGISTER. BITS 8,9 AND 6-1 ARE TESTED TO
;SET INDIVIDUALLY AS WELL AS COLLECTIVELY WITHOUT DESTROYING
;ANY PREVIOUS DATA PATTERN

```

STARS

;*****

BEGIN.TEST

T11::

```

MOV #CSPAT,R3 ;GET BEGINNING OF LIST
BGNSEG ;**** START SEGMENT **** TRAP C$BSEG
MOV #CRDY,@RLCS ;INSURE GO IS THERE
MOV (R3),GDDAT ;SET UP EXPECTED RLCS...
BIS #CRDY,GDDAT ;...IN GDDAT.
BIS (R3),@RLCS ; ** BIT SET PATTERN IN RLCS
BIT #DERR,@RLCS ;IF ERROR BIT SET THEN...
BEQ 2$ ;...EXPECT IT ON THE READ BACK.
BIS #ERR!DERR,GDDAT
MOV @RLCS,BDDAT ;READ RLCS TO CHECK "BIS"
BIC #DRDY,BDDAT ;CLEAR OUT DRIVE READY
CMP BDDAT,GDDAT ;DID BIS WORK?
BEQ 3$ ;BRANCH IF OKAY
DFERR EM61,ERR2 ;WRONG DATA IN RLCS ON BIS. TRAP C$ERDF
;WORD 110
;WORD EM61
;WORD ERR2
3$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE
;WORD 10000$-.
4$: MOV #1776,@RLCS ;SET ALL SETTABLE BITS
MOV #1776,GDDAT ;SET UP EXPECT DATA IN...
BIC (R3),GDDAT ;...GDDAT
BIC (R3),@RLCS ; ** XCT BIC ON RLCS.
BIT #DERR,@RLCS ;IF DRIVE ERROR BIT SET...
BEQ 5$ ;...EXPECT IT SET WHEN WE READ.
BIS #ERR!DERR,GDDAT
MOV @RLCS,BDDAT ;MOVE RLCS TO BDDAT FOR COMPARE
BIC #DRDY,BDDAT ;CLEAR DRIVE READY
CMP BDDAT,GDDAT ;DID "BIC" WORK PROPERLY
BEQ 6$ ;BRANCH IF OKAY
DFERR EM62,ERR2 ;WRONG DATA IN RLCS ON BIC. TRAP C$ERDF
;WORD 111
;WORD EM62
;WORD ERR2
6$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE
;WORD 10000$-.
TST (R3)+ ;GET NEXT PATTERN
CMP R3,#CSEND ;AT END OF LIST
BNE 1$ ;NO, GO BACK WITH NEXT PATTERN

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 19-1
11 -- BIS AND BIC OF RLCS.

SEQ 0060

1461 013736
(3) 013736
(3) 013736 104405
1462 013740
(3) 013740
(3) 013740 104401

ENDSEG

ENDTST

10000\$: TRAP C#ESEG

L10025: TRAP C#ETST

.SBTTL 12 -- BIS AND BIC OF RLBA.

1464
1465
1466 013742
(2)
1467
1468
1469
1470
1471 013742
(2)
1472 013742
(4) 013742
1473 013742 012703 003672
1474 013746
(3) 013746 104404
1475 013750 005077 167352
1476 013754 011337 003432
1477 013760 023727 003402 000000
1478 013766 001003
1479 013770 042737 000001 003432
1480 013776 051377 167324
1481 014002 017737 167320 003434
1482 014010 023737 003434 003432
1483 014016 001404
1484 014020
(5) 014020 104455
(6) 014022 000160
(6) 014024 033007
(6) 014026 026272
1485 014030
(3) 014030 104410
(3) 014032 000070
1486 014034 012777 177776 167264
1487 014042 012737 177776 003432
1488 014050 041337 003432
1489 014054 041377 167246
1490 014060 017737 167242 003434
1491 014066 023737 003434 003432
1492 014074 001404
1493 014076
(5) 014076 104455
(6) 014100 000161
(6) 014102 033055
(6) 014104 026272
1494 014106
(3) 014106 104410
(3) 014110 000012
1495 014112 005723
1496 014114 020327 004100
1497 014120 001313
1498 014122
(3) 014122
(3) 014122 104405
1499 014124
(3) 014124
(3) 014124 104401

STARS

;TEST THAT THE "BIS" AND "BIC" INSTRUCTIONS WILL WORK ON THE
;BUS ADDRESS REGISTER. BITS 15-0 ARE LOADED, ONLY BITS 15-1
;ARE EXPECTED BACK. FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
;GROWING 0, AND SHIFTING 0.

STARS

BEGIN.TEST

T12::

MOV #BEGPAT,R3 ;GET START OF LIST
BGNSEG ;**** START SEGMENT **** TRAP C\$BSEG
1\$: CLR @RLBA ;CLEAR "BA"
MOV (R3),GDDAT ;SET EXPECTED
CMP RL11,#RL11 ; RL11 ??
BNE 2\$; NO.
BIC #1,GDDAT ;BIT 0 CAN'T SET IN RLBA (UNIBUS)
2\$: BIS (R3),@RLBA ; XCT BIS RLBA WITH PATTERN
MOV @RLBA,BDDAT ;READ "BA"
CMP BDDAT,GDDAT ;DID RLBA LOAD PROPERLY?
BEQ 3\$;BRANCH IF YES
DFERR EM63,ERR2 ;WRONG DATA IN RLBA ON BIS. TRAP C\$ERDF
;WORD 112
;WORD EM63
;WORD ERR2
3\$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C\$ESCAPE
;WORD 10000\$-.
4\$: MOV #-2,@RLBA ;SET RLBA TO ALL 1'S (BIT 0=0)
MOV #-2,GDDAT ;SET UP EXPECTED RESULTS
BIC (R3),GDDAT ;IN GDDAT
BIC (R3),@RLBA ; XCT BIC RLBA
MOV @RLBA,BDDAT ;READ RLBA
CMP BDDAT,GDDAT ;BIC WORK OKAY?
BEQ 5\$;IF YES BRANCH
DFERR EM64,ERR2 ;WRONG DATA IN RLBA ON BIC. TRAP C\$ERDF
;WORD 113
;WORD EM64
;WORD ERR2
5\$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA:LOE). TRAP C\$ESCAPE
;WORD 10000\$-.
TST (R3)+ ;GET NEXT PATTERN
CMP R3,#ENDPAT ;HAVE WE COMPLETED LIST
BNE 1\$;NO, GO BACK FOR NEXT
ENDSEG
10000\$: TRAP C\$ESEG
L10026: TRAP C\$ETST
ENDTST

.SBTTL 13 -- BIS AND BIC OF RLDA.

1501
1502
1503 014126
(2)
1504
1505
1506
1507 014126
(2)
1508 014126
(4) 014126
1509 014126 012703 003672
1510 014132
(3) 014132 104404
1511 014134 005077 167170
1512 014140 011337 003432
1513 014144 051377 167160
1514 014150 017737 167154 003434
1515 014156 023737 003434 003432
1516 014164 001404
1517 014166
(5) 014166 104455
(6) 014170 000162
(6) 014172 033125
(6) 014174 026272
1518 014176
(3) 014176 104410
(3) 014200 000070
1519 014202 012777 177777 167120
1520 014210 012737 177777 003432
1521 014216 041337 003432
1522 014222 041377 167102
1523 014226 017737 167076 003434
1524 014234 023737 003432 003434
1525 014242 001404
1526 014244
(5) 014244 104455
(6) 014246 000163
(6) 014250 033173
(6) 014252 026272
1527 014254
(3) 014254 104410
(3) 014256 000012
1528 014260 005723
1529 014262 020327 004100
1530 014266 001322
1531 014270
(3) 014270
(3) 014270 104405
1532 014272
(3) 014272
(3) 014272 104401

STARS

;TEST THAT THE "BIS" AND "BIC" INSTRUCTIONS WILL WORK ON THE DISK
;ADDRESS REGISTER. BITS 15-0 ARE TESTED WITH 4 PATTERNS, GROWING 1,
;SHIFTING 1, GROWING 0, AND SHIFTING 0.

STARS

BEGIN.TEST

T13::

MOV #BEGPAT,R3 ;GET START OF LIST
BGNSEG ; **** START SEGMENT **** TRAP C\$BSEG
1\$: CLR @RLDA ;CLEAR "DA"
MOV (R3),GDDAT ;SET EXPECTED
BIS (R3),@RLDA ; XCT BIS RLDA
MOV @RLDA,BDDAT ;READ RLDA
CMP BDDAT,GDDAT ;IS RLDA CORRECT
BEQ 2\$;IF OKAY BRANCH
DFERR EM65,ERR2 ;WRONG DATA IN RLDA ON BIS. TRAP C\$ERDF
;WORD 114
;WORD EM65
;WORD ERR2
2\$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C\$ESCAPE
;WORD 10000\$-.
3\$: MOV #-1,@RLDA ;SET RLDA TO ALL 1'S
MOV #-1,GDDAT ;SET EXPECTED DATA
BIC (R3),GDDAT ;SET EXPECTED DATA
BIC (R3),@RLDA ; XCT "BIC" RLDA
MOV @RLDA,BDDAT ;READ RLDA
CMP GDDAT,BDDAT ;DID "BIC" WORK?
BEQ 4\$;IF IT DID BRANCH
DFERR EM66,ERR2 ;WRONG DATA IN RLDA ON BIC. TRAP C\$ERDF
;WORD 115
;WORD EM66
;WORD ERR2
4\$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C\$ESCAPE
;WORD 10000\$-.
TST (R3)+ ;GET NEXT PATTERN
CMP R3,#ENDPAT ;DONE?
BNE 1\$;NO GO BACK
ENDSEG 10000\$: TRAP C\$ESEG
ENDTST L10027: TRAP C\$ETST

.SBTTL 14 -- BIS AND BIC OF RLBAE (RLV12 ONLY).

```

1534
1535
1536 014274
(2)
1537
1538
1539
1540
1541 014274
(2)
1542 014274
(4) 014274
1543 014274 023727 003402 000003
1544 014302 001070
1545 014304 012703 003672
1546 014310
(3) 014310 104404
1547 014312 005077 167016 1$: CLR @RLBAE ;CLEAR "BAE"
1548 014316 011337 003432 MOV (R3),GDDAT ;SET EXPECTED...
1549 014322 042737 177700 003432 BIC #+C77,GDDAT ;...ONLY <5:0> ARE VALID.
1550 014330 051377 167000 BIS (R3),@RLBAE ; XCT BIS RLBAE WITH PATTERN
1551 014334 017737 166774 003434 MOV @RLBAE,BDDAT ;READ "BAE"
1552 014342 023737 003434 003432 CMP BDDAT,GDDAT ;DID IT LOAD PROPERLY?
1553 014350 001404 BEQ 2$ ;BRANCH IF YES
1554 014352
(5) 014352 104455 DFERR EM67,ERR2 ;WRONG DATA IN RLBAE ON BIS. TRAP C$ERDF
(6) 014354 000164 .WORD 116
(6) 014356 033243 .WORD EM67
(6) 014360 026272 .WORD ERR2
1555 014362 2$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE
(3) 014362 104410 .WORD 10000$-.
(3) 014364 000076
1556 014366 012777 177777 166740 3$: MOV #-1,@RLBAE ;SET RLBAE TO ALL 1'S
1557 014374 012737 177777 003432 MOV #-1,GDDAT ;SET UP EXPECTED RESULTS...
1558 014402 041337 003432 BIC (R3),GDDAT ;...IN GDDAT...
1559 014406 042737 177700 003432 BIC #+C77,GDDAT ;...<5:0> ONLY.
1560 014414 041377 166714 BIC (R3),@RLBAE ; XCT BIC RLBAE
1561 014420 017737 166710 003434 MOV @RLBAE,BDDAT ;READ RLBAE
1562 014426 023737 003434 003432 CMP BDDAT,GDDAT ;BIC WORK OKAY?
1563 014434 001404 BEQ 4$ ;IF YES BRANCH
1564 014436
(5) 014436 104455 DFERR EM68,ERR2 ;WRONG DATA IN RLBAE ON BIC. TRAP C$ERDF
(6) 014440 000165 .WORD 117
(6) 014442 033312 .WORD EM68
(6) 014444 026272 .WORD ERR2
1565 014446 4$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE
(3) 014446 104410 .WORD 10000$-.
(3) 014450 000012
1566 014452 005723 TST (R3)+ ;GET NEXT PATTERN
1567 014454 020327 004100 CMP R3,#ENDPAT ;HAVE WE COMPLETED LIST
1568 014460 001314 BNE 1$ ;NO, GO BACK FOR NEXT
1569 014462
(3) 014462 ENDSEG 10000$: TRAP C$ESEG
(3) 014462 104405
1570 014464 5$: ENDTST L10030:
(3) 014464

```


M5

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 22-1
14 -- BIS AND BIC OF RLBAE (RLV12 ONLY).

SEQ 0064

(3) 014464 104401

TRAP C#ETST

.SBTTL 15 -- UNIQUENESS OF RLCS.

1572
1573
1574 014466
(2)
1575
1576
1577
1578
1579
1580
1581
1582 014466
(2)
1583 014466
(4) 014466
1584 014466 012777 177776 166632
1585 014474 012777 177777 166626
1586 014502 023727 003402 000003
1587 014510 001003
1588 014512 012777 000077 166614
1589 014520 012777 000200 166576 1\$:
1590 014526 012737 177776 003432
1591 014534 017737 166566 003434
1592 014542 023737 003432 003434
1593 014550 001405
1594 014552
(5) 014552 104455
(6) 014554 000166
(6) 014556 033544
(6) 014560 026272
1595 014562
(3) 014562 104406
1596 014564 012737 177777 003432 2\$:
1597 014572 017737 166532 003434
1598 014600 023737 003432 003434
1599 014606 001405
1600 014610
(5) 014610 104455
(6) 014612 000167
(6) 014614 033577
(6) 014616 026272
1601 014620
(3) 014620 104406
1602 014622 023727 003402 000003 3\$:
1603 014630 001016
1604 014632 012737 000074 003432
1605 014640 017737 166470 003434
1606 014646 023737 003432 003434
1607 014654 001404
1608 014656
(5) 014656 104455
(6) 014660 000170
(6) 014662 033632
(6) 014664 026272
1609 014666
(3) 014666

STARS

;TEST THE UNIQUENESS OF THE CONTROL AND STATUS
;REGISTER. THE RLBA AND RLDA ARE PRELOADED WITH
;177776 AND 177777 RESPECTIVELY. THE RLCS IS THEN
;LOADED TO INSURE THAT NEITHER THE RLBA OR RLDA
;ARE MODIFIED BY THE WRITING OF THE RLCS.
; IF RLV12, CHECK THE BAE ALSO, NOTING THAT CSR<5:4> SHOULD
; BE MIRRORED IN BAE<1:0>.

STARS

BEGIN.TEST

MOV #-2,@RLBA ;SET RLBA TO ALL 1'S
MOV #-1,@RLDA ;SET RLDA TO ALL 1'S
CMP RLTP,#RLV12X
BNE 1\$
MOV #77,@RLBAE ; SET BAE TO ALL 1'S.
MOV #CRDY,@RLCS ;WRITE RLCS
MOV #-2,GDDAT ;SET UP EXPECTED BA.
MOV @RLBA,BDDAT ;READ RLBA
CMP GDDAT,BDDAT
BEQ 2\$; BR IF BA UNCHANGED.
DFERR EM72,ERR2 ;CS MODIFIED BA

TRAP C\$ERDF
.WORD 118
.WORD EM72
.WORD ERR2

CKLOOP

MOV #-1,GDDAT ;SET UP EXPECTED DA.
MOV @RLDA,BDDAT ;READ DA
CMP GDDAT,BDDAT
BEQ 3\$; BR IF DA UNCHANGED.
DFERR EM73,ERR2 ;CS MODIFIED DA

TRAP C\$CLP1

CKLOOP

CMP RLTP,#RLV12X
BNE 4\$
MOV #74,GDDAT ; SET EXPECTED BAE.
MOV @RLBAE,BDDAT ; READ IT.
CMP GDDAT,BDDAT
BEQ 4\$; BR IF BAE IS RIGHT.
DFERR EM73A,ERR2 ; BAE WRONG AFTER WRITING CS.

TRAP C\$ERDF
.WORD 119
.WORD EM73
.WORD ERR2

TRAP C\$CLP1

4\$: ENDTST

L10031:

B6

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 23-1
15 -- UNIQUENESS OF RLCS.

SEQ 0066

(3) 014666 104401

TRAP C4ETST

1611
1612
1613 014670
(2)
1614
1615
1616
1617
1618
1619 014670
(2)
1620 014670
(4) 014670
1621 014670 012737 000200 003432
1622 014676 032777 040000 166420
1623 014704 001403
1624 014706 052737 140000 003432
1625 014714 013777 003432 166402 1\$:
1626 014722 012777 177777 166400
1627 014730 023727 003402 000003
1628 014736 001003
1629 014740 012777 000074 166366
1630 014746 005077 166354 2\$:
1631 014752 017737 166346 003434
1632 014760 042737 000001 003434
1633 014766 023737 003432 003434
1634 014774 001405
1635 014776 104455
(5) 014776 104455
(6) 015000 000171
(6) 015002 033671
(6) 015004 026272
1636 015006
(3) 015006 104406
1637 015010 012737 177777 003432 3\$:
1638 015016 017737 166306 003434
1639 015024 023737 003432 003434
1640 015032 001405
1641 015034
(5) 015034 104455
(6) 015036 000172
(6) 015040 033723
(6) 015042 026272
1642 015044
(3) 015044 104406
1643 015046 023727 003402 000003 4\$:
1644 015054 001016
1645 015056 012737 000074 003432
1646 015064 017737 166244 003434
1647 015072 023737 003432 003434
1648 015100 001404
1649 015102
(5) 015102 104455
(6) 015104 000173
(6) 015106 033755
(6) 015110 026272

.SBTTL 16 -- UNIQUENESS OF RLBA.

STARS

;TEST THE UNIQUENESS OF THE BUS ADDRESS REGISTER. THE
;RLCS AND RLDA ARE LOADED WITH XXX20X AND 177777
;RESPECTIVELY. THE RLBA IS THEN WRITTEN TO INSURE
;THAT NEITHER THE RLCS OR RLDA ARE MODIFIED.
; IF RLV12, CHECK THAT BAE IS UNAFFECTED ALSO.

STARS

BEGIN.TEST

T16::

MOV #CRDY,GDDAT ;CONTROLLER READY
BIT #DERR,@RLCS ;IF DRIVE ERROR IS SET...
BEQ 1\$
BIS #ERR!DERR,GDDAT ;...EXPECT IT BACK.
MOV GDDAT,@RLCS ;LOAD RLCS
MOV #-1,@RLDA ;LOAD RLDA
CMP RLTP,#RLV12X
BNE 2\$
MOV #74,@RLBAE ;LOAD RLBAE.
CLR @RLBA ;WRITE TO RLBA
MOV @RLCS,BDDAT ;READ RLCS
BIC #DRDY,BDDAT ;IGNORE DRIVE READY
CMP GDDAT,BDDAT
BEQ 3\$;BR IF CS UNCHANGED.
DFERR EM74,ERR2 ;BA MODIFIED CS

TRAP C\$ERDF
.WORD 121
.WORD EM74
.WORD ERR2

CKLOOP

MOV #-1,GDDAT ;SET EXPECTED DA.
MOV @RLDA,BDDAT ;READ IT.
CMP GDDAT,BDDAT
BEQ 4\$;BR IF DA UNCHANGED.
DFERR EM75,ERR2 ;BA MODIFIED DA

TRAP C\$CLP1

CKLOOP

CMP RLTP,#RLV12X
BNE 5\$
MOV #74,GDDAT ;SET EXPECTED BAE.
MOV @RLBAE,BDDAT ;READ IT.
CMP GDDAT,BDDAT
BEQ 5\$;BR IF BAE UNCHANGED.
DFERR EM75A,ERR2 ;BA MODIFIED BAE

TRAP C\$ERDF
.WORD 123
.WORD EM75A
.WORD ERR2

CVRLB80 -- RLV12 DISKLESS.
CVRLB8.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 24-1
16 -- UNIQUENESS OF RLBA.

D6

SEQ 0068

1650 015112
(3) 015112
(3) 015112 104401

54: ENDTST

L10032: TRAP C4ETST

.SBTTL 17 -- UNIQUENESS OF RLDA.

STARS

;TEST THE UNIQUENESS OF THE DISK ADDRESS REGISTER. THE RLCS
;AND RLBA ARE LOADED WITH XXX20X AND 177776 RESPECTIVELY.
;THE RLDA IS THEN WRITTEN TO INSURE THAT
;NEITHER THE RLCS OR THE RLBA ARE MODIFIED.
; IF RLV12, CHECK THAT RLBAE IS UNAFFECTED ALSO.

STARS

BEGIN.TEST

T17::

1652
1653
1654 015114
(2)
1655
1656
1657
1658
1659
1660 015114
(2)
1661 015114
(4) 015114
1662 015114 012737 000200 003432
1663 015122 032777 040000 166174
1664 015130 001403
1665 015132 052737 140000 003432
1666 015140 013777 003432 166156 1\$:
1667 015146 012777 177776 166152
1668 015154 023727 003402 000003
1669 015162 001003
1670 015164 012777 000074 166142
1671 015172 005077 166132 2\$:
1672 015176 017737 166122 003434
1673 015204 042737 000001 003434
1674 015212 023737 003432 003434
1675 015220 001405
1676 015222
(5) 015222 104455
(6) 015224 000174
(6) 015226 034011
(6) 015230 026272
1677 015232
(3) 015232 104406
1678 015234 012737 177776 003432 3\$:
1679 015242 017737 166060 003434
1680 015250 023737 003432 003434
1681 015256 001405
1682 015260
(5) 015260 104455
(6) 015262 000175
(6) 015264 034044
(6) 015266 026272
1683 015270
(3) 015270 104406
1684 015272 023727 003402 000003 4\$:
1685 015300 001016
1686 015302 012737 000074 003432
1687 015310 017737 166020 003434
1688 015316 023737 003432 003434
1689 015324 001404
1690 015326
(5) 015326 104455
(6) 015330 000176
(6) 015332 034077
(6) 015334 026272

MOV #CRDY,GDDAT ;CONTROLLER READY
BIT #DERR,@RLCS ;IF DRIVE ERROR IS SET...
BEQ 1\$
BIS #ERR!DERR,GDDAT ;...EXPECT IT BACK.
MOV GDDAT,@RLCS ;LOAD CS
MOV #-2,@RLBA ;LOAD BA WITH ALL 1'S
CMP RLTY,#RLV12X
BNE 2\$
MOV #74,@RLBAE ; LOAD BAE.
CLR @RLDA ; WRITE TO RLDA.
MOV @RLCS,BDDAT ;READ RLCS
BIC #DRDY,BDDAT ;IGNORE DRIVE READY
CMP GDDAT,BDDAT
BEQ 3\$; BR IF CS UNCHANGED.
DFERR EM76,ERR2 ;DA MODIFIED CS

TRAP C\$ERDF
.WORD 124
.WORD EM76
.WORD ERR2

CKLOOP

MOV #-2,GDDAT ; SET EXPECTED BA.
MOV @RLBA,BDDAT ; READ IT.
CMP GDDAT,BDDAT
BEQ 4\$; BR IF BA UNCHANGED.
DFERR EM77,ERR2 ;DA MODIFIED BA

TRAP C\$CLP1

CKLOOP

CMP RLTY,#RLV12X
BNE 5\$
MOV #74,GDDAT ; SET EXPECTED BAE.
MOV @RLBAE,BDDAT ; READ IT.
CMP GDDAT,BDDAT
BEQ 5\$; BR IF BAE UNCHANGED.
DFERR EM77A,ERR2 ; DA MODIFIED BAE.

TRAP C\$ERDF
.WORD 126
.WORD EM77A
.WORD ERR2

F6

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 25-1
17 -- UNIQUENESS OF RLDA.

SEQ 0070

1691 015336
(3) 015336
(3) 015336 104401

54: ENDTST

L10033: TRAP C4ETST

1693
1694
1695 015340
(2)
1696
1697
1698
1699
1700
1701 015340
(2)
1702 015340
(4) 015340
1703 015340 012737 000200 003432
1704 015346 032777 040000 165750
1705 015354 001403
1706 015356 052737 140000 003432
1707 015364 013777 003432 165732 1\$:
1708 015372 012777 177776 165726
1709 015400 012777 177777 165722
1710 015406 023727 003402 000003
1711 015414 001003
1712 015416 012777 000074 165710
1713 015424 005077 165702 2\$:
1714 015430 017737 165670 003434
1715 015436 042737 000001 003434
1716 015444 023737 003432 003434
1717 015452 001405
1718 015454
(5) 015454 104455
(6) 015456 000177
(6) 015460 032405
(6) 015462 026272
1719 015464
(3) 015464 104406
1720 015466 012737 177776 003432 3\$:
1721 015474 017737 165626 003434
1722 015502 023737 003432 003434
1723 015510 001405
1724 015512
(5) 015512 104455
(6) 015514 000200
(6) 015516 032440
(6) 015520 026272
1725 015522
(3) 015522 104406
1726 015524 012737 177777 003432 4\$:
1727 015532 017737 165572 003434
1728 015540 023737 003432 003434
1729 015546 001405
1730 015550
(5) 015550 104455
(6) 015552 000201
(6) 015554 032473
(6) 015556 026272
1731 015560

.SBTTL 18 -- UNIQUENESS OF RLMP.

STARS

;;*****
;TEST THE UNIQUENESS OF THE MULTI-PURPOSE REGISTER
;WE WILL WRITE THE RLCS, RLBA, AND THE RLDA, THEN THE
;RLMP IS WRITTEN. WE THEN GO BACK AN VERIFY THE CONTENTS
;OF THE RLCS, RLBA, RLDA.
; IF RLV12, INCLUDE THE RLBAE ALSO.

STARS

;;*****

BEGIN.TEST

T18::

MOV #CRDY,GDDAT ;CONTROLLER READY
BIT #DERR,@RLCS ;IF DRIVE ERROR IS SET...
BEQ 1\$
BIS #ERR!DERR,GDDAT ;...EXPECT IT BACK LATER.
MOV GDDAT,@RLCS ;LOAD CS
MOV #-2,@RLBA ;LOAD BA WITH ALL 1'S
MOV #-1,@RLDA ;LOAD RLDA
CMP RLTP,#RLV12X
BNE 2\$
MOV #74,@RLBAE ;LOAD RLBAE.
CLR @RLMP ;WRITE TO RLMP
MOV @RLCS,BDDAT ;READ RLCS
BIC #DRDY,BDDAT ;IGNORE DRIVE READY
CMP GDDAT,BDDAT
BEQ 3\$; BR IF CS OK.
DFERR EM44,ERR2 ;MP MODIFIED CS

TRAP C\$ERDF
.WORD 127
.WORD EM44
.WORD ERR2

CKLOOP

TRAP C\$CLP1

MOV #-2,GDDAT ; SET EXPECTED BA.
MOV @RLBA,BDDAT ; READ IT.
CMP GDDAT,BDDAT
BEQ 4\$; BR IF BA OK.
DFERR EM45,ERR2 ;MP MODIFIED BA

TRAP C\$ERDF
.WORD 128
.WORD EM45
.WORD ERR2

CKLOOP

TRAP C\$CLP1

MOV #-1,GDDAT ; SET EXPECTED DA.
MOV @RLDA,BDDAT ; READ IT.
CMP GDDAT,BDDAT
BEQ 5\$; BR IF DA OK.
DFERR EM46,ERR2 ;MP MODIFIED DA

TRAP C\$ERDF
.WORD 129
.WORD EM46
.WORD ERR2

CKLOOP

```

(3) 015560 104406
1732 015562 023727 003402 000003 5$: CMP RL TYP,#RLV12X
1733 015570 001016 BNE 6$
1734 015572 012737 000074 003432 MOV #74,GDDAT ; SET EXPECTED BAE.
1735 015600 017737 165530 003434 MOV @RLBAE,BDDAT ; READ IT.
1736 015606 023737 003432 003434 CMP GDDAT,BDDAT
1737 015614 001404 BEQ 6$ ; BR IF BAE OK.
1738 015616 DFERR EM47,ERR2 ; MP MODIFIED BAE.
(5) 015616 104455
(6) 015620 000202
(6) 015622 032526
(6) 015624 026272
1739 015626 6$: ENDTST
(3) 015626
(3) 015626 104401

```

TRAP C\$CLP1

TRAP C\$ERDF
 .WORD 130
 .WORD EM47
 .WORD ERR2

L10034:
 TRAP C\$ETST

.SBTTL 19 -- UNIQUENESS OF RLBAE (RLV12 ONLY).

1741
1742
1743 015630
(2)
1744
1745
1746
1747
1748 015630
(2)
1749 015630
(4) 015630
1750 015630 023727 003402 000003
1751 015636 001071
1752 015640 012737 000200 003432
1753 015646 032777 040000 165450
1754 015654 001403
1755 015656 052737 140000 003432
1756 015664 013777 003432 165432 1\$:
1757 015672 005077 165430
1758 015676 005077 165426
1759 015702 012777 000077 165424
1760 015710 052737 000060 003432
1761 015716 017737 165402 003434
1762 015724 042737 000001 003434
1763 015732 023737 003432 003434
1764 015740 001405
1765 015742
(5) 015742 104455
(6) 015744 000203
(6) 015746 032562
(6) 015750 026272
1766 015752
(3) 015752 104406
1767 015754 005037 003432 2\$:
1768 015760 017737 165342 003434
1769 015766 001405
1770 015770
(5) 015770 104455
(6) 015772 000204
(6) 015774 032621
(6) 015776 026272
1771 016000
(3) 016000 104406
1772 016002 017737 165322 003434 3\$:
1773 016010 001404
1774 016012
(5) 016012 104455
(6) 016014 000205
(6) 016016 032645
(6) 016020 026272
1775 016022 4\$:
(3) 016022
(3) 016022 104401

STARS

;TEST THAT WRITING TO THE RLBAE HAS NO AFFECT ON
;THE RLBA AND RLDA REGISTERS. THE RLCS REGISTER WILL
;BE AFFECTED, BUT ONLY IN THE EXTENDED ADDRESS BITS <5:4>,
;WHICH SHOULD MIRROR RLBAE<1:0>.
STARS

STARS

BEGIN.TEST

T19::

CMP RL TYP,#RLV12X ; RLV12 WITH BAE ??
BNE 4\$; EXIT IF NOT.
MOV #CRDY,GDDAT ;CONTROLLER READY
BIT #DERR,@RLCS ;IF DRIVE ERROR IS SET...
BEQ 1\$
BIS #ERR!DERR,GDDAT ;...EXPECT IT BACK LATER.
MOV GDDAT,@RLCS ;LOAD CS, NOTE THAT <5:4> = 0
CLR @RLBA ; 0 => BA
CLR @RLDA ; 0 => DA
MOV #77,@RLBAE ; WRITE TO RLBAE.
BIS #60,GDDAT ; SET EXPECTED CS.
MOV @RLCS,BDDAT ; READ IT.
BIC #DRDY,BDDAT ; IGNORE DRIVE READY BIT.
CMP GDDAT,BDDAT
BEQ 2\$; BR IF CS IS RIGHT.
DFERR EM50,ERR2 ; CS WRONG AFTER WRITING BAE.

TRAP C\$ERDF
.WORD 131
.WORD EM50
.WORD ERR2

CKLOOP

TRAP C\$CLP1

CLR GDDAT ; EXPECT ZERO ON THE REST.
MOV @RLBA,BDDAT ; READ BA
BEQ 3\$; BR IF UNAFFECTED.
DFERR EM51,ERR2 ; BAE MODIFIED BA.

TRAP C\$ERDF
.WORD 132
.WORD EM51
.WORD ERR2

CKLOOP

TRAP C\$CLP1

MOV @RLDA,BDDAT ; READ DA.
BEQ 4\$; BR IF UNAFFECTED.
DFERR EM52,ERR2 ; BAE MODIFIED DA.

TRAP C\$ERDF
.WORD 133
.WORD EM52
.WORD ERR2

ENDTST

L10035:

TRAP C\$ETST

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 28
20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).

SEQ 0074

```

1777 .SBTTL 20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).
1778
1779 016024 STARS
(2) ;*****
1780 ; TEST FUNCTION CODE 0.
1781 ; IF RL11 -- NOP. EXPECT CS<13:10> CLEAR, OTHER REGISTERS UNAFFECTED.
1782 ; IF RLVXX -- MAINT. SETUP IN NORMAL FASHION, BUT ONLY CHECK FOR
1783 ; FUNCTION COMPLETE (NO ERRORS AND DA = INITIAL DA+6).
1784 ; WE'LL CHECK THE REST OF THE MAINT FUNCTION LATER.
1785 016024 STARS
(2) ;*****
1786 016024 BEGIN.TEST
(4) 016024 T20::
1787 ;
1788 ; INITIAL SETUP ASSUMES THAT WE HAVE AN RLV.
1789 ; IF NOT -- NOT TO WORRY -- IT SHOULD JUST NO-OP.
1790 ; ALL WE CARE ABOUT IS THAT WE GET NO ERRORS (OR HUNG RL).
1791 ;
1792 016024 004537 010760 1$: JSR R5,CALCRC ; MAKE A CRC EVEN THO WE WON'T CHECK IT.
1793 016030 000000 .WORD 0 ; THIS WILL BE THE INITIAL DA WORD.
1794 016032 004537 010254 JSR R5,LDFUN ; LOAD UP THE FUNCTION.
1795 016036 000000 177001 0, -511. ; FUNCTION 0, WC -511.
1796 016042 004537 010704 JSR R5,WTCRDY ; WAIT FOR CONTROLLER.
1797 016046 023727 003402 000000 CMP RLTP,#RL11 ; NOW ARE WE RL11 ??
1798 016054 001045 BNE 6$ ;CHECK "MAINT" FINAL STATE IF NOT.
1799 ;
1800 ; RL11 -- CHECK FOR ERRORS AFTER THE NO-OP.
1801 ;
1802 ;LWLOi 2$: CLR B.MP ; MPR WILL BE 0.
1803 016056 013737 003350 003432 2$: MOV B.CS,GDDAT ; EXPECT "NO-OP" SHOULD HAVE...
1804 016064 042737 036000 003432 BIC #036000,GDDAT ; ...CLEARED ANY ERROR BITS <13:10>...
1805 016072 013737 003364 003434 MOV E.CS,BDDAT ; ...THAT MIGHT HAVE BEEN ON.
1806 016100 023737 003432 003434 CMP GDDAT,BDDAT ; IS THAT WHAT WE HAVE ??
1807 016106 001405 BEQ 3$ ; YES.
1808 016110 DFERR EM14,ERR2 ; CS WRONG AFTER NOP.
(5) 016110 104455 TRAP C$ERDF
(6) 016112 000206 .WORD 134
(6) 016114 031457 .WORD EM14
(6) 016116 026272 .WORD ERR2
1809 016120 CKLOOP TRAP C$CLP1
(3) 016120 104406
1810 016122 023737 003366 003352 3$: CMP E.BA,B.BA ; BA UNALTERED ??
1811 016130 001010 BNE 4$
1812 016132 023737 003370 003354 CMP E.DA,B.DA ; DA UNALTERED ??
1813 016140 001004 BNE 4$
1814 016142 023737 003372 003356 CMP E.MP,B.MP ; MP UNALTERED ??
1815 016150 001405 BEQ 5$ ; EXIT IF ALL THREE WERE OK.
1816 016152 4$: DFERR EM14A,ERRO ; REGISTERS ALTERED BY NOP.
(5) 016152 104455 TRAP C$ERDF
(6) 016154 000207 .WORD 135
(6) 016156 031512 .WORD EM14A
(6) 016160 026236 .WORD ERRO
1817 016162 CKLOOP TRAP C$CLP1
(3) 016162 104406
1818 016164 5$: EXIT TST ; RL NOP TEST DONE.
(3) 016164 104432 TRAP C$EXIT

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 28-1
20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).

SEQ 0075

```

(3) 016166 000042 .WORD L10036-
1819 ;
1820 ; RLV11 OR 12 -- CHECK SEQUENCER STATE AFTER INTERNAL DIAGNOSTIC.
1821 ;
1822 016170 013737 003370 003434 6$: MOV E,DA,BDDAT ; GET FINAL SEQUENCER STATE.
1823 016176 023727 003434 000006 CMP BDDAT,#6 ; FINAL STATE RIGHT ??
1824 016204 001004 BNE 7$ ; NO, ERROR.
1825 016206 032737 002000 003364 BIT #OPI,E.CS ; YES, OPI SHOULD BE CLEAR.
1826 016214 001405 BEQ 8$ ; IF IT IS, WE'RE FAT !!
1827 016216 7$: DFERR EM14B,ERR3 ; MAINT FAILURE, DA = SEQUENCER STATE.
(5) 016216 104455 TRAP C$ERDF
(6) 016220 000210 .WORD 136
(6) 016222 031554 .WORD EM14B
(6) 016224 026326 .WORD ERR3
1828 016226 CKLOOP TRAP C$CLP1
(3) 016226 104406
1829 016230 8$: ENDTST
(3) 016230 L10036: TRAP C$ETST
(3) 016230 104401

```


.SBTTL 21 -- TEST INTERRUPT ON FUNCTION (0) COMPLETE.

1831
1832
1833 016232
(2)
1834
1835
1836
1837
1838
1839
1840 016232
(2)
1841 016232
(4) 016232
1842 016232 005037 010700
1843 016236
(3) 016236 012700 000000
(3) 016242 104441
1844 016244 004537 010760
1845 016250 000000
1846 016252 004537 010254
1847 016256 000100 177001
1848 016262 004537 010704
1849 016266 005737 010700
1850 016272 001006
1851 016274 005037 003424
1852 016300
(5) 016300 104455
(6) 016302 000211
(6) 016304 031616
(6) 016306 026442
1853 016310
(3) 016310 012700 000340
(3) 016314 104441
1854 016316
(3) 016316
(3) 016316 104401

STARS

```

;*****
;TEST THAT THE RL CAN INTERRUPT THE CPU.
;WE'LL SET CPU PRIORITY AT 0, AND EXECUTE NOP/MAINT, AS BEFORE.
;THE INTERRUPT SERVICE DOES NOTHING BUT SET A FLAG.
;WE'LL WAIT 800 MSEC FOR THAT FLAG, BEFORE DECLARING AN ERROR.
;WRONG VECTORS UNDER 1000 ARE TRAPPED BY THE DRS.
;WRONG VECTORS ABOVE 1000 ARE BIG TROUBLE !!!!

```

STARS

;*****

BEGIN.TEST

```

CLR INTFLG ;CLEAR INTERRUPT FLAG
SETPRI #PRI00 ;SET PSW TO 0

```

T21::

```

MOV #PRI00,RO
TRAP C$SPRI

```

```

JSR R5,CALCRC ; AGAIN, CRC WON'T BE CHECKED.

```

.WORD 0

JSR R5,LDFUN

; LOAD UP AND EXECUTE...

INTEN!0, -511.

; NOP/MAINT AND INTERRUPT.

JSR R5,WTCRDY

;WAIT FOR CONTROLLER READY.

TST INTFLG

;DID INTERRUPT OCCUR ??

BNE 1\$

; BR IF SO.

CLR TMO

; CPU LEVEL FOR ERROR MSG.

DFERR EM15,ERR7

; INTERRUPT NOT REC'D.

```

TRAP C$ERDF
.WORD 137
.WORD EM15
.WORD ERR7

```

1\$: SETPRI #PRI07

```

MOV #PRI07,RO
TRAP C$SPRI

```

ENDTST

L10037:

```

TRAP C$ETST

```

```

1856                                     .SBTTL 22 -- TEST INTERRUPT PRIORITY BR LEVEL.
1857
1858 016320                               STARS
(2)                                     ;*****
1859                                     ;TEST THAT PRIORITY GIVEN IS ACTUAL PRIORITY OF CONTROLLER.
1860                                     ;WE ALREADY KNOW THAT THE CONTROLLER CAN INTERRUPT. WE'LL START
1861                                     ;AT CPU LEVEL 7 AND WORK DOWN 'TIL THE INTERRUPT COMES IN.
1862                                     ; AS BEFORE, WE'RE USING THE NOP/MAINT FUNCTION.
1863                                     ; NOTE THAT RLV11 IS FIXED AT BR4, RL11 AND RLV12 ARE PROGRAMABLE
1864                                     ; AND DEFAULT TO BR5. LOC "BPRIOR" IS PROPERLY SET AT INIT TIME.
1865 016320                               STARS
(2)                                     ;*****
1866 016320                               BEGIN.TEST
(4) 016320                               T22::
1867 016320 012737 000340 003434          MOV #PRI07,BDDAT ;SET UP INITIAL OF 7
1868 016326 012737 000007 003424          MOV #7,TMPO ; A COPY FOR ERROR MSG.
1869 016334 013737 003340 003432          MOV BPRIOR,GDDAT ;GET GIVEN PRIORITY
1870 016342                                     BGNSEG ; **** START SEGMENT ****
(3) 016342 104404                                     TRAP C$BSEG
1871 016344                                     1$: SETPRI BDDAT ; SET CURRENT PRIORITY...
(3) 016344 013700 003434                                     MOV BDDAT,RO
(3) 016350 104441                                     TRAP C$SPRI
1872 016352 005037 010700                                     CLR INTFLG ;...AND CLEAR INTERRUPT FLAG.
1873 016356 004537 010760                                     JSR R5,CALCRC ; JUST LIKE BEFORE.
1874 016362 000000                                     .WORD 0
1875 016364 004537 010254                                     JSR R5,LDGUN ; LOAD AND EXECUTE...
1876 016370 000100 177001                                     INTEN!0, -511. ;...NOP/MAINT AND INTERRUPT.
1877 016374 004537 010704                                     JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY.
1878 016400 023737 003434 003432          CMP BDDAT,GDDAT ;SHOULD IT INTERRUPT
1879 016406 002004                                     BGE 2$ ; BR IF NOT.
1880 016410 005737 010700                                     TST INTFLG ; DID IT INTERRUPT ??
1881 016414 001011                                     BNE 4$ ; PROCEED IF SO.
1882 016416 000403                                     BR 3$ ; NO, ERROR.
1883 016420 005737 010700                                     2$: TST INTFLG ; DID IT INTERRUPT ??
1884 016424 001405                                     BEQ 4$ ; PROCEED IF NOT.
1885 016426                                     3$: DFERR EM16,ERR7 ; INTERRUPT PRIORITY FAILURE.
(5) 016426 104455                                     TRAP C$ERDF
(6) 016430 000212                                     .WORD 138
(6) 016432 031660                                     .WORD EM16
(6) 016434 026442                                     .WORD ERR7
1886 016436                                     CKLOOP
(3) 016436 104406                                     TRAP C$CLP1
1887 016440 162737 000040 003434 4$: SUB #40,BDDAT ;LOWER CPU PRIORITY...
1888 016446 005337 003424          DEC TMPO ;...TO NEXT LEVEL.
1889 016452 100334          BPL 1$
1890 016454          ENDSEG
(3) 016454                                     10000$: TRAP C$ESEG
(3) 016454 104405
1891 016456          SETPRI #PRI07
(3) 016456 012700 000340          MOV #PRI07,RO
(3) 016462 104441          TRAP C$SPRI
1892 016464          ENDTST
(3) 016464                                     L10040: TRAP C$ETST
(3) 016464 104401

```

CVRLB80 -- RLV12 DISKLESS.
CVRLB8.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 31
23 -- RLV11/12 MAINTENANCE, FORCED OPI (WC <> 511.)

SEQ 0078

.SBTTL 23 -- RLV11/12 MAINTENANCE, FORCED OPI (WC <> 511.)

STARS

;EXECUTE MAINTENANCE MODE WITH AN INCORRECT INITIAL WORD COUNT.
;FIRST PASS WC < 510. AND SECOND PASS WC > 511. BOTH CASES SHOULD
;FORCE OPI ERRORS. THE TEST SHOULD FORCE COMPOSITE ERROR(BIT 15),
;HEADER NOT FOUND(BIT12) AND OPI(BIT 10). DRIVE ERROR IS IGNORED.
STARS

BEGIN.TEST

T23::

;* NOTE: IF CONTROLLER TYPE IS RL11, YOU HAVE END-PASS AT THIS POINT.
;* ALL REMAINING TESTS ARE UNIQUE TO RLV11 AND/OR RLV12.
;*

1\$: JSR R5,CALCRC ;DO CRC CALCULATION FOR...
.WORD 0 ;...INITIAL DA OF ZERO.
MOV #-509.,3\$; 1ST WC < 510.
MOVB #'<,EM27X ; ADJUST ERROR TEXT.
MOVB #'0,EM27X+4
2\$: JSR R5,LDFUN ;PERFORM MAINT FUNCTION
MAINT
3\$: -509. ; TEST WORD COUNT.
JSR R5,WTCRDY
MOV #ERR!HNF!OPI,RO ;EXPECT AT LEAST THESE ERRORS.
BIC E.CS,RO
BEQ 4\$; EXIT IF WE'RE OK.
JSR R5,GETERR ; OTHERWISE, GET THEM ALL...
NOP ;...NONE IS EQUALLY BAD NEWS.
DFERR EM27,ERR10 ; STATUS INCORRECT, WC <> 511.

TRAP C\$ERDF
.WORD 139
.WORD EM27
.WORD ERR10

CKLOCP

TRAP C\$CLP1

4\$: CMP 3\$, #-512. ; 2ND PASS ??
BEQ 5\$; WE'RE DONE IF SO.
MOV #-512.,3\$; OTHERWISE, SET WC > 511.
MOVB #'>,EM27X ; ADJUST ERROR TEXT.
MOVB #'1,EM27X+4
BR 2\$; AND GO 'ROUND.
5\$: ENDTST

L10041:

TRAP C\$ETST

1894
1895
1896 016466
(2)
1897
1898
1899
1900
1901 016466
(2)
1902 016466
(4) 016466
1903
1904
1905
1906
1907 016466 004537 010760
1908 016472 000000
1909 016474 012737 177003 016524
1910 016502 112737 000074 032323
1911 016510 112737 000060 032327
1912 016516 004537 010254
1913 016522 000000
1914 016524 177003
1915 016526 004537 010704
1916 016532 012700 112000
1917 016536 043700 003364
1918 016542 001410
1919 016544 004537 007554
1920 016550 000240
1921 016552
(5) 016552 104455
(6) 016554 000213
(6) 016556 032223
(6) 016560 026472
1922 016562
(3) 016562 104406
1923 016564 023727 016524 177000 4\$:
1924 016572 001412
1925 016574 012737 177000 016524
1926 016602 112737 000076 032323
1927 016610 112737 000061 032327
1928 016616 000737
1929 016620
(3) 016620
(3) 016620 104401

1932
1933
1934 016622
(2)
1935
1936
1937
1938 016622
(2)
1939 016622
(4) 016622
1940 016622 004537 010760
1941 016626 000000
1942 016630
(3) 016630 012700 000000
(3) 016634 104441
1943 016636 005037 010700
1944 016642 004537 010254
1945 016646 000100 177003
1946 016652 004537 010704
1947 016656 005737 010700
1948 016662 001005
1949 016664
(5) 016664 104455
(6) 016666 000214
(6) 016670 031713
(6) 016672 026236
1950 016674
(3) 016674 104406
1951 016676
(3) 016676 012700 000340
(3) 016702 104441
1952 016704
(3) 016704
(3) 016704 104401

.SBTTL 24 -- RLV11/12 MAINTENANCE, FORCED OPI INTERRUPT.

STARS

```

;*****
;PERFORM TEST OF INTERRUPT BY ISSUING RLV11 MAINTENANCE FUNCTION
;WITH A WORD COUNT OF -509. TO FORCE AN OPI ERROR.
;CHECK THAT INTERRUPT OCCURS, REPORT ERROR IF NOT.

```

STARS

;*****

BEGIN.TEST

T24::

```

1$: JSR R5,CALCRC ;CALCULATE CRC FOR...
      .WORD 0 ;...INITIAL DA OF ZERO.
      SETPRI #PRI00 ;SET PRIORITY TO ZERO

      CLR INTFLG ;CLEAR INT. FLAG
      JSR R5,LDFUN
      MAINT!INTEN, -509. ; INVALID WC TO FORCE OPI ERROR.
      JSR R5,WTCRDY ;WAIT FOR READY
      TST INTFLG ; DID IT INTERRUPT ??
      BNE 4$ ; YES, THAT'S ALL.
      DFERR EM17,ERRO ; NO INTERRUPT RECEIVED.

      TRAP C$ERDF
      .WORD 140
      .WORD EM17
      .WORD ERRO

      CKLOOP
      TRAP C$CLP1

4$: SETPRI #PRI07
      MOV #PRI07,RO
      TRAP C$SPRI

      ENDTST
      L10042:
      TRAP C$ETST

```

```

1954
1955
1956 016706
  (2)
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966 016706
  (2)
1967 016706
  (4) 016706
1968 016706 005737 003400
1969 016712 100447
1970 016714 004537 010760
1971 016720 000000
1972 016722 012704 000002
1973 016726 004537 010254
1974 016732 000000 177003
1975 016736 004537 010704
1976 016742 012737 001440 003434
1977 016750 166637 177774 003434
1978 016756 032777 002000 164340
1979 016764 001005
1980 016766
  (5) 016766 104455
  (6) 016770 000215
  (6) 016772 032332
  (6) 016774 026236
1981 016776 000415
1982 017000 023737 003434 003450 3$:
1983 017006 003004
1984 017010 023737 003434 003446
1985 017016 002005
1986 017020 077436 4$:
1987 017022
  (5) 017022 104455
  (6) 017024 000216
  (6) 017026 032360
  (6) 017030 026522
1988 017032
  (3) 017032
  (3) 017032 104401

```

```

.SBTTL 25 -- RLV11/12 MAINTENANCE, OPI TIMING TEST.

STARS
;*****
;NOTE: THE SOFT TIMER EMPLOYED HERE IS CALIBRATED FOR LSI'S ONLY.
; IF THE CPU TYPE IS UNKNOWN (NOT LSI), BYPASS THIS TEST.
;
;PERFORM RLV11 MAINTENANCE FUNCTION (0) IN FLAG MODE.
;FORCE AN OPI TIMEOUT BY SETTING AN INVALID WORD COUNT.
;MEASURE THE TIME UNTIL THE ERROR FLAG SETS AND COMPARE THAT
;TIME AGAINST THE SPEC LIMITS (155 TO 650 MSEC).
; NOTE: SINCE THE TIMING LOOP IS SO GROSS, WE'LL GIVE IT
; A SECOND CHANCE BEFORE WE DECLARE AN ERROR.
STARS
;*****
      BEGIN.TEST
                                     T25::
      TST      CPUTYP      ; LSI CPU ???
      BMI      5$          ; IF NOT, DON'T EVEN TRY !!!
1$:     JSR     R5,CALCRC   ; MAKE CRC FOR ZERO.
      .WORD   0
      MOV     #2,R4
2$:     JSR     R5,LDFUN   ;PERFORM MAINT. FUNCTION
      MAINT, -509.        ; MAINT WITH INVALID WC.
      JSR     R5,WTCRDY   ; WAIT FOR DONE.
      MOV     #800.,BDDAT ; CALCULATE OPI TIME...
      SUB     -4(SP),BDDAT ; ...IN BDDAT.
      BIT     #OPI,@RLCS ; OPI ERROR ??
      BNE     3$          ; YES, CHECK TIMING.
      DFERR   EM31,ERRO   ; OPI FLAG NOT RECEIVED.
                                     TRAP   C$ERDF
                                     .WORD  141
                                     .WORD  EM31
                                     .WORD  ERRO
      BR      5$
3$:     CMP     BDDAT,OPIMX
      BGT     4$
      CMP     BDDAT,OPIMN
      BGE     5$          ; OK IF TIME WITHIN LIMITS.
4$:     SOB     R4,2$     ; TRY ONCE MORE.
      DFERR   EM32,ERR11 ;OPI TIMING INCORRECT
                                     TRAP   C$ERDF
                                     .WORD  142
                                     .WORD  EM32
                                     .WORD  ERR11
5$:     ENDTST
                                     L10043:
                                     TRAP   C$ETST

```

.SBTTL 26 -- RLV11/12 MAINTENANCE, FIFO DMA AND CRC CHECK.

1990
1991
1992 017034
(2)
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004 017034
(2)
2005 017034
(4) 017034
2006
2007
2008
2009 017034
(3) 017034
(3) 017034 104402
2010 017036 012704 004202
2011 017042 016437 000072 017054
2012 017050 004537 010422
2013 017054 000000
2014 017056 012437 017066
2015 017062 004537 010760
2016 017066 000000
2017 017070 004737 017204
2018 017074 005714
2019 017076 001361
2020 017100
(3) 017100
(3) 017100 104403
2021
2022
2023
2024 017102
(3) 017102
(3) 017102 104402
2025 017104 012704 004202
2026 017110 016437 000072 017122
2027 017116 004537 010426
2028 017122 000000
2029 017124 012437 017134
2030 017130 004537 010760
2031 017134 000000
2032 017136 004737 017204
2033 017142 005714
2034 017144 001361
2035 017146
(3) 017146

STARS

;PERFORM RLV11/12 MAINTENANCE FUNCTION 0 IN FLAG MODE AND CHECK
;FOR PROPER INCREMENT OF THE DA AND BA REGISTERS. CHECK THE SERIAL
;WRITE/READ DATA PATHS BY READING OUT OF THE FIFO VIA THE MP REGISTER
;THE CRC OF DA+3 AND THE CRC OF CRC OF DA+4 AND COMPARING WITH EXPECTED
;RESULTS. CHECK THE TRANSFER OF 255 WORDS FROM BUF1 MEMORY THROUGH THE
;FIFO INTO BUF2 MEMORY FOR PROPER DATA.
;CHECK THE PREVIOUSLY WRITTEN DATA IN THE LAST WORD+1 OF BUF2 FOR
;A VALUE:123456 TO INSURE THAT THE TRANSFER WAS NOT MORE THAN 255 WORDS.
;DMA 1 -- USES 28 DATA PATTERNS IN "PATCRC" AND "PATDAT".
;DMA 2 -- USES COMPLIMENT OF THOSE DATA PATTERNS.
;DMA 3 -- USES A RANDOM 256 WORD PATTERN.

STARS

BEGIN.TE ,

T26::

; SUBTEST 1 -- 28 DIFFERENT DATA PATTERNS.

; DMA1: BGNSUB

T26.1:

TRAP C\$BSUB

1\$: MOV #PATCRC,R4 ;SET TABLE POINTER.
MOV 72(R4),2\$; GET TEST DATA...
JSR R5,SETPAT ;...AND FILL THE BUFFER.
2\$: .WORD 0
MOV (R4)+,3\$; GET DA TEST WORD...
JSR R5,CALCRC ;...AND CALCULATE CRC'S.
3\$: .WORD 0
JSR PC,EXECUT ; XCT MAINT MODE AND CHECK RESULTS.
TST (R4) ; LAST PATTERN DONE ??
BNE 1\$; LOOP IF NOT.
ENDSUB

L10045:

TRAP C\$ESUB

; SUBTEST 2 -- THE COMPLIMENT OF THE ABOVE 28 PATTERNS.

; DMA2: BGNSUB

T26.2:

TRAP C\$BSUB

1\$: MOV #PATCRC,R4 ;RESET TABLE POINTER.
MOV 72(R4),2\$; FILL WITH COMPLIMENT DATA.
JSR R5,SETCMP
2\$: .WORD 0
MOV (R4)+,3\$; CALCULATE CRC'S.
JSR R5,CALCRC
3\$: .WORD 0
JSR PC,EXECUT ; XCT
TST (R4) ; LAST ONE DONE ??
BNE 1\$; LOOP IF NOT.
ENDSUB

L10046:


```

(3) 017146 104403                                TRAP  C$ESUB
2036
2037      ; SUBTEST 3 -- ONE RANDOM DATA PATTERN.
2038      ;
2039 017150                                DMA3:  BGNSUB                                T26.3:
(3) 017150
(3) 017150 104402                                TRAP  C$BSUB
2040 017152 004537 010516                        JSR    R5,SETRAN      ;FILL BUFFER WITH RANDOM DATA.
2041 017156 013737 003456 017170                MOV    TEMLO,1$      ; USE TEMLO FOR THE CRC WORD.
2042 017164 004537 010760                        JSR    R5,CALCRC     ; CALCULATE CRC'S.
2043 017170 000000                                1$:    .WORD 0
2044 017172 004737 017204                        JSR    PC,EXECUT     ; XCT
2045 017176                                ENDSUB
(3) 017176                                L10047:
(3) 017176 104403                                TRAP  C$ESUB
2046 017200                                EXIT  TST             ; ALL DONE.
(3) 017200 104432                                TRAP  C$EXIT
(3) 017202 000530                                .WORD L10044-.
2047      ;
2048      ; THIS SUBROUTINE EXECUTES THE FUNCTION AND VERIFIES RESULTS.
2049      ;
2050 017204                                EXECUT: BGNSEG      ; *** SET LOOP SEGMENT ***
(3) 017204 104404                                TRAP  C$BSEG
2051 017206 004537 010254                        JSR    R5,LDFUN      ;PERFORM MAINT. FUNCTION
2052 017212 000000 177001                        MAINT, -511.
2053 017216 004537 010704                        JSR    R5,WTCRDY     ;WAIT FOR READY
2054 017222 004537 007554                        JSR    R5,GETERR     ;CHECK CONTROLLER FOR ERRORS
2055 017226 000405                                BR     11$           ; BR IF NONE.
2056 017230                                DFERR  EM98,ERR6     ;UNEXPECTED CONTROLLER ERRORS.
(5) 017230 104455                                TRAP  C$ERDF
(6) 017232 000217                                .WORD 143
(6) 017234 034421                                .WORD EM98
(6) 017236 026412                                .WORD ERR6
2057 017240                                CKLOOP
(3) 017240 104406                                TRAP  C$CLP1
2058 017242 032737 022000 003364 11$:          BIT    #OPI!NXM,E.CS ; OPI OR NXM ??
2059 017250 001402                                BEQ    12$           ; PROCEED IF NOT.
2060 017252                                EXIT  SEG             ; YES, DON'T BOTHER WITH ANY MORE.
(3) 017252 104432                                TRAP  C$EXIT
(3) 017254 000452                                .WORD 10000$-.
2061 017256 012737 006364 003432 12$:          MOV    #BUFEND-2,GDDAT
2062 017264 013737 003366 003434                MOV    E.BA,BDDAT
2063 017272 023737 003432 003434                CMP    GDDAT,BDDAT  ;FINAL BA CORRECT ??
2064 017300 001405                                BEQ    1$
2065 017302                                DFERR  EM10,ERR4     ;FINAL BA INCORRECT.
(5) 017302 104455                                TRAP  C$ERDF
(6) 017304 000220                                .WORD 144
(6) 017306 031355                                .WORD EM10
(6) 017310 026356                                .WORD ERR4
2066 017312                                CKLOOP
(3) 017312 104406                                TRAP  C$CLP1
2067 017314 013737 003354 003432 1$:          MOV    B.DA,GDDAT   ;GET BEFORE DA REGISTER
2068 017322 013737 003370 003434                MOV    E.DA,BDDAT
2069 017330 005037 003424                        CLR    TMPO
2070 017334 113737 003354 003424                MOVB   B.DA,TMPO
2071 017342 062737 000006 003424                ADD    #6,TMPO      ;+6 TO DA LOW BYTE

```



```

(6) 017610 012746 000005
(3) 017614 010600
(4) 017616 104415
(4) 017620 062706 000014
2104 017624          CKLOOP
(3) 017624 104406
2105 017626 005722      7$: TST (R2)+ ;INCREMENT BUFFER POINTERS.
2106 017630 005723      TST (R3)+
2107 017632 005201      INC R1 ;AND THE WORD COUNT.
2108 017634 020127 000377  CMP R1,#255. ; DONE ALL WORDS ??
2109 017640 002724      BLT 5$ ; NOT YET, CONTINUE.
2110 017642 005737 003430  TST TMP2 ; YES, ANY ERRORS LOGGED ??
2111 017646 001412      BEQ 8$ ;NO.
2112 017650          PRINTB #FRMT15,TMP2 ;YES, PRINT SUMMARY.
(8) 017650 013746 003430      MOV TMP2,-(SP)
(7) 017654 012746 030172      MOV #FRMT15,-(SP)
(6) 017660 012746 000002      MOV #2,-(SP)
(3) 017664 010600          MOV SP,R0
(4) 017666 104414          TRAP C$PNTB
(4) 017670 062706 000006      ADD #6,SP
2113 017674 012737 123456 003432 8$: MOV #123456,GDDAT ;EXPECTED DATA IN LAST WORD+1
2114 017702 011237 003434      MOV (R2),BDDAT ;GET LAST WORD+1 FROM BUF2
2115 017706 023737 003432 003434  CMP GDDAT,BDDAT
2116 017714 001404          BEQ 9$
2117 017716          DFERR EM23,ERR4 ; LAST+1 INCORRECT.
(5) 017716 104455          TRAP C$ERDF
(6) 017720 000225          .WORD 149
(6) 017722 032155          .WORD EM23
(6) 017724 026356          .WORD ERR4
2118 017726          9$: ENDSEG
(3) 017726          10000$:
(3) 017726 104405          TRAP C$ESEG
2119 017730 000207          RTS PC ; RETURN.
2120
2121 017732          10$: ENDTST
(3) 017732          L10044:
(3) 017732 104401          TRAP C$ETST

```


2123
2124
2125 017734
(2)
2126
2127
2128
2129
2130
2131
2132 017734
(2)
2133 017734
(4) 017734
2134
2135
2136
2137 017734
(3) 017734
(3) 017734 104402
2138 017736 005001
2139 017740 012702 000400
2140 017744 012703 004366
2141 017750 010123
2142 017752 005201
2143 017754 005302
2144 017756 001374
2145 017760 004737 020026
2146 017764
(3) 017764
(3) 017764 104403
2147
2148
2149
2150 017766
(3) 017766
(3) 017766 104402
2151 017770 012701 177777
2152 017774 012702 000400
2153 020000 012703 004366
2154 020004 010123
2155 020006 005301
2156 020010 005302
2157 020012 001374
2158 020014 004737 020026
2159 020020
(3) 020020
(3) 020020 104403
2160 020022
(3) 020022 104432
(3) 020024 000332
2161
2162
2163
2164 020026
(3) 020026 104404

```
.SBTTL 27 -- RLV11/12 MAINTENANCE, FIFO ADDRESSING.

STARS
;*****
;TEST THAT FIFO OPERATES CORRECTLY.  STORE ADDRESS PATTERN
;IN BUF1 (0-255) THAT CONTAINS A UNIQUE PATTERN IN EACH LOCATION.
;PERFORM MAINTENANCE FUNCTION AND TEST BUF2 FOR PROPER FIFO
;ADDRESSING.  NOTE THAT CRC'S ARE NOT CHECKED IN THIS TEST.
;REPEAT A SECOND TIME USING A COMPLIMENT ADDRESS PATTERN.
; ALSO USE AND TEST FOR INTERRUPT ON MAINT DONE.
STARS
;*****
      BEGIN.TEST
                                T27::

;
; SUBTEST 1 -- FIFO ADDRESS PATTERN.
;
FIF01:  BGNSUB
                                T27.1:  TRAP    C$BSUB
                                L10051:
                                TRAP    C$ESUB

;
; SUBTEST 2 -- FIFO COMPLIMENT ADDRESS PATTERN.
;
FIF02:  BGNSUB
                                T27.2:  TRAP    C$BSUB
                                L10052:
                                TRAP    C$ESUB

      EXIT  TST                    ; ALL DONE.
                                TRAP    C$EXIT
                                .WORD  L10050-.

;
; THIS SUBROUTINE EXECUTES THE MAINT FUNCTION AND TESTS RESULTS.
;
XFIFO:  BGNSEG                    ;*** LOOP SEGMENT ***
                                TRAP    C$BSEG
```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 36-1
27 -- RLV11/12 MAINTENANCE, FIFO ADDRESSING.

SEQ 0086

2165	020030	012702	000400			MOV	#256.,R2		;SETUP TO CLEAR BUF2		
2166	020034	012703	005366			MOV	#BUF2,R3				
2167	020040	005023			1\$:	CLR	(R3)+				
2168	020042	005302				DEC	R2				
2169	020044	001375				BNE	1\$				
2170	020046	005037	010700			CLR	INTFLG		;CLEAR INT. FLAG		
2171	020052					SETPRI	#PRI00				
(3)	020052	012700	000000							MOV	#PRI00,R0
(3)	020056	104441								TRAP	C\$SPRI
2172	020060	004537	010254			JSR	R5, LDFUN		;LOAD FUNCTION		
2173	020064	000100	177001				MAINT!INTEN, -511.				
2174	020070	004537	010704			JSR	R5, WTCRDY		;WAIT FOR READY		
2175	020074	004537	007554			JSR	R5, GETERR		;CHECK CONTROLLER FOR ERRORS		
2176	020100	000405				BR	2\$; BR IF NONE.		
2177	020102					DFERR	EM98,ERR6		;UNEXPECTED CONTROLLER ERRORS.		
(5)	020102	104455								TRAP	C\$ERDF
(6)	020104	000226								.WORD	150
(6)	020106	034421								.WORD	EM98
(6)	020110	026412								.WORD	ERR6
2178	020112					CKLOOP					
(3)	020112	104406								TRAP	C\$CLP1
2179	020114	032737	022000	003364	2\$:	BIT	#OPI!NXM,E.CS		; OPI OR NXM ??		
2180	020122	001110				BNE	8\$; QUIT IF SO.		
2181	020124	005737	010700			TST	INTFLG		;CHECK FOR INTERRUPT		
2182	020130	001005				BNE	4\$				
2183	020132					DFERR	EM15,ERRO		; DIDN'T INTERRUPT.		
(5)	020132	104455								TRAP	C\$ERDF
(6)	020134	000227								.WORD	151
(6)	020136	031616								.WORD	EM15
(6)	020140	026236								.WORD	ERRO
2184	020142					CKLOOP					
(3)	020142	104406								TRAP	C\$CLP1
2185	020144	005037	003430		4\$:	CLR	TMP2		;CLEAR BAD WORD COUNTER		
2186	020150	012703	004366			MOV	#BUF1,R3		;GOOD DATA STORED IN BUF1		
2187	020154	012702	005366			MOV	#BUF2,R2		;DATA BUFFER WRITTEN INTO BY MAINT.		
2188	020160	005001				CLR	R1		; WORD COUNT.		
2189	020162	011337	003432		5\$:	MOV	(R3),GDDAT		;EXPECTED DATA		
2190	020166	011237	003434			MOV	(R2),BDDAT		;GET DATA FROM BUFFER		
2191	020172	023737	003432	003434		CMP	GDDAT,BDDAT				
2192	020200	001436				BEQ	7\$;BR IF DATA IS RIGHT.		
2193	020202	010237	003424			MOV	R2,TMP0		; GET ERROR ADDRESS...		
2194	020206	010137	003426			MOV	R1,TMP1		;...AND WORD NUMBER.		
2195	020212	005737	003430			TST	TMP2		; 1ST ERROR ENCOUNTERED ??		
2196	020216	001004				BNE	6\$;NO, SKIP THE ERROR HEADER.		
2197	020220					DFERR	EM22,ERRO		;ERROR MESSAGE ON 1ST ERROR...		
(5)	020220	104455								TRAP	C\$ERDF
(6)	020222	000230								.WORD	152
(6)	020224	032111								.WORD	EM22
(6)	020226	026236								.WORD	ERRO
2198	020230	005237	003430		6\$:	INC	TMP2		;...FOLLOWED BY FAILING DATA.		
2199	020234					PRINTX	#FRMT14,TMP1,TMP0,GDDAT,BDDAT				
(11)	020234	013746	003434							MOV	BDDAT,-(SP)
(10)	020240	013746	003432							MOV	GDDAT,-(SP)
(9)	020244	013746	003424							MOV	TMP0,-(SP)
(8)	020250	013746	003426							MOV	TMP1,-(SP)
(7)	020254	012746	030103							MOV	#FRMT14,-(SP)

.SBTTL 28 -- RLV11/12 MAINTENANCE, BANK 7 SELECT AND NEXM TEST.

STARS

```

*****
;* FOR RLV12:
;* TEST THAT BBS7 WILL SELECT THE I/O PAGE AND THAT ACCESS TO
;* LOCATION 0 IN THAT PAGE WILL GENERATE NXM AND OPI ERRORS.
;* NOTE -- IF BANK 7 IS NOT PROPERLY SELECTED, 1000 BYTES
;* STARTING AT XXXX1000 WILL PROBABLY GET CRUNCHED !!!!
;* FOR RLV11:
;* RLV11 DOESN'T ASSERT BBS7, SO WE'LL THIS TEST
;* IF PHYSICAL MEMORY SIZE IS 124K OR GREATER.

```

STARS

BEGIN.TEST

T28::

```

2215
2216
2217 020360
(2)
2218
2219
2220
2221
2222
2223
2224
2225
2226 020360
(2)
2227 020360
(4) 020360
2228 020360 023727 003402 000001
2229 020366 003006
2230 020370 023727 011276 000174
2231 020376 002406
2232 020400
(3) 020400 104432
(3) 020402 000150
2233 020404 112737 000054 034164 1$:
2234 020412 000402
2235 020414 105037 034164 2$:
2236 020420 004537 010760
2237 020424 000000
2238 020426 012737 160000 010416
2239 020434 012700 000003
2240 020440 023727 011304 000026
2241 020446 001002
2242 020450 012700 000077
2243 020454 010037 010420 3$:
2244 020460 004537 010254
2245 020464 000000 177001
2246 020470 004537 010704
2247 020474 012700 120000
2248 020500 043700 003364
2249 020504 001415
2250
2251 020506 004537 007554
2252 020512 000240
2253 020514 013737 010420 003426
2254 020522 013737 010416 003430
2255 020530
(5) 020530 104455
(6) 020532 000231
(6) 020534 034133
(6) 020536 026552
2256
2257 020540 012737 004366 010416 4$:
2258 020546 005037 010420
2259 020552
(3) 020552
(3) 020552 104401

```

```

CMP RLTP,#RLV11
BGT 1$
CMP .MSIZE,#124.
BLT 2$
EXIT TST

```

```

; EXECUTE ALWAYS ON RLV12...
; ...AND ON RLV11 IF < 124KW.

```

```

TRAP C$EXIT
.WORD L10053-.

```

```

MOV #',,EMBOX
SKP2
CLR B EMBOX
JSR R5,CALCRC
.WORD 0
MOV #160000,BA16
MOV #3,RO
CMP .ABUSW,#22.
BNE 3$
MOV #77,RO
MOV RO,BA22
JSR R5,LDLUN
MAINT,-511.
JSR R5,WTCRDY
MOV #ERR!NXM,RO
BIC E.CS,RO
BEQ 4$

```

```

; ADJUST ERROR TEXT FOR RLV12...
; ...OR FOR RLV11.
; DO CRC FOR INITIAL...
; ...DA OF 0.
; POINT TO LOC 0 IN LAST 4K.
; SET AN 18 BIT...
; ...OR 22 BIT EXTENSION.
; XCT MAINTENANCE...
; ...WITH VALID WORD COUNT.
; EXPECT AT LEAST THESE ERRORS.
; EXIT IF WE'RE OK.

```

```

JSR R5,GETERR
NOP
MOV BA22,TMP1
MOV BA16,TMP2
DFERR EM80,ERR12

```

```

; OTHERWISE, GET WHAT'S THERE.
; AND GET PHYSICAL ADDRESS.
; BBS7 AND/OR NXM FAILURE.

```

```

TRAP C$ERDF
.WORD 153
.WORD EM80
.WORD ERR12

```

```

MOV #BUF1,BA16
CLR BA22
ENDTST

```

; RESET BA16 AND 22.

```

L10053: TRAP C$ETST

```

.SBTTL 29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.

2261
2262
2263 020554
(2)

2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275

2276 020554
(2)

2277 172350
(4)

2278 020554
(4) 020554

2279 020554 005037 010672

2280 020560 005737 177572

2281 020564 000240

2282 020566 005737 010672

2283 020572 001402

2284 020574
(3) 020574 104432

(3) 020576 000550

2285 020600 012737 007400 021324

2286 020606 023727 011304 000026

2287 020614 001012

2288 020616 023727 003402 000003

2289 020624 001006

2290 020626 012737 000020 172516

2291 020634 012737 177400 021324

2292 020642 005037 010420

2293 020646 012737 177000 010416

2294 020654 012737 001770 172350

2295 020662 004537 010516

2296 020666
(3) 020666 104404

2297 020670 012701 004366

2298 020674 012702 100000

2299 020700 012703 001000

010672
177572
010672
001402
104432
000550
007400 021324
011304 000026
003402 000003
172516
177400 021324
010420
010416
172350
010516
104404
004366
100000
001000
020734 000004
010672
177572
000240
000240
077304
000405
010672
020746
000002
177572

STARS

;* BYPASS THIS TEST IF NO KT SUPPORT.
;* OTHERWISE, MEMORY HAS ALREADY BEEN SIZED.
;*PAR'S 0 TO 6 ARE MAPPED TO THE 1ST 28K, AND PAR7 TO THE I/O PAGE.
;*
;*NOW ACCESS EXTENDED MEMORY IN 4K INCREMENTS. STORE A RANDOM DATA
;*PATTERN IN BUF1 (RELOCATED). EXECUTE MAINTENANCE MODE, AND CHECK
;*BUF2 (RELOCATED) FOR CORRECT DATA TRANSFER. ALSO CHECK THAT THE FINAL
;*BUS ADDRESS INCREMENTED CORRECTLY (OVERFLOWING INTO THE BAE).
;*WHEN THE END OF PHYSICAL MEMORY IS REACHED, A BUS-ERROR WILL OCCUR
;*DURING THE BUFFER SET-UP PHASE AND WE'LL LOOK FOR A NXM ERROR AT
;*THE SAME PHYSICAL ADDRESS. CONTINUE UNTIL WE REACH THE I/O PAGE
;*AT 760000 (18 BIT), OR 17760000 (22 BIT).

STARS

PAR4= KIPAR0+10 ; USE KIPAR4 FOR ACCESSING.
BEGIN.TEST

CLR TRPFLG ; CLEAR TRAP FLAG. T29::
TST MMRO ; KT AVAILABLE ??
240
TST TRPFLG
BEQ EXMTST ; PROCEED IF SO...
1\$: EXIT TST ;...EXIT OTHERWISE.

EXMTST: MOV #7400,IOPAG ; ASSUME 18 BITS, SET LIMIT AT 124K...
CMP .ABUSW,#22. ; 22 BITS AVAILABLE ??
1\$ BNE ; BR IF NOT.
CMP RLTY,#RLV12X ; CAN WE HANDLE THEM ALL ??
1\$ BNE ; BR IF NOT.
MOV #20,MMR3 ; YES, SET KT IN 22 BIT MODE...
MOV #177400,IOPAG ;...AND RAISE LIMIT TO 2044K.

1\$: CLR BA22
MOV #177000,BA16 ; SET PHYSICAL ADDRESS AT 32K-256 WORDS.
MOV #1770,PAR4 ; SET PAR4 TO THE SAME POINT.
EXMLUP: JSR R5,SETRAN ; RANDOM DATA TO BUF1, ZERO BUF2.
BGNSEG

TRAP C\$EXIT L10054-..
TRAP C\$BSEG
MOV #BUF1,R1 ; SETUP TO COPY FROM BUF1...
MOV #100000,R2 ;...TO (PAR4)+0
MOV #512,R3
MOV #22\$,ERRVEC ; SOONER OR LATER WE'LL TRAP.

2\$: CLR TRPFLG
INC MMRO ;***** KT ON *****
MOV (R1)+,(R2)+ ; COPY BUFFERS VIA PAR4. *
240,240 ; *
SOB R3,2\$; LOOP 'TIL DONE. *
BR 3\$; *
22\$: MOV R2,TRPFLG ; NXM TRAP, SAVE ADDRESS... *
MOV #3\$,(SP) ; *
RTI ;...AND CONTINUE. *
3\$: CLR MMRO ;***** KT OFF *****

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 38-1
29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.

SEQ 0090

```

2311 020752 012737 010670 000004      MOV    #TRAP4,ERRVEC    ; RESET TRAP.
2312 020760 004537 010760              JSR    R5,CALCRC       ; JUST TO BE CONSISTENT.
2313 020764 000000 000000              0
2314 020766 004537 010254              JSR    R5,LDFUN
2315 020772 000000 177001      MAINT, -511.          ; EXECUTE MAINT MODE DMA TO EXT MEM.
2316 020776 004537 010704              JSR    R5,WTCRDY
2317 021002 023727 003402 000003      CMP    RLTP,#RLV12X   ; BAE IN USE ??
2318 021010 001411 000000              BEQ    33$            ; BR IF SO.
2319 021012 013700 003364              MOV    E.CS,R0        ; NO, GET EA<17:16> FROM FINAL CSR.
2320 021016 006200 000000              ASR    R0
2321 021020 006200 000000              ASR    R0
2322 021022 006200 000000              ASR    R0
2323 021024 006200 000000              ASR    R0
2324 021026 042700 177774              BIC    #C3,R0
2325 021032 000402 000000              SKP2
2326 021034 013700 003376      33$:  MOV    E.BAE,R0        ; GET EA<21:16> FROM BAE.
2327 021040 010037 003426              MOV    R0,TMP1        ; SAVE FINAL PA IN CASE...
2328 021044 013737 003366      003430  MOV    E.BA,TMP2      ; ...WE HAVE A STATUS ERROR.
2329 021052 005737 010672              TST    TRPFLG         ; NOW, ARE WE EXPECTING NXM ??
2330 021056 001415 000000              BEQ    4$             ; BR IF NOT.
2331 021060 012700 120000              MOV    #ERR!NXM,R0   ; YES, DO WE HAVE IT ??
2332 021064 043700 003364              BIC    E.CS,R0
2333 021070 001501 000000              BEQ    8$             ; BR IF SO.
2334 021076 004537 007554              JSR    R5,GETERR     ; NO, GET WHATEVER'S THERE.
2335 021076 000240 000000              NOP
2336 021100 000240 000000      DFERR  EM90,ERR12    ; EXTENDED NXM STATUS WRONG.
(5) 021100 104455 000000              TRAP   C$ERDF
(6) 021102 000232 000000              .WORD 154
(6) 021104 034221 000000              .WORD EM90
(6) 021106 026552 000000              .WORD ERR12
2337 021110 000471 000000              BR     8$
2338
2339 021112 004537 007554      4$:  JSR    R5,GETERR    ; ANY OTHER STATUS ERRORS ??
2340 021116 000405 000000              BR     44$           ; BR IF NOT.
2341 021120 000405 000000      DFERR  EM91,ERR13    ; STATUS INCORRECT ON EXT MEMORY ACCESS.
(5) 021120 104455 000000              TRAP   C$ERDF
(6) 021122 000233 000000              .WORD 155
(6) 021124 034275 000000              .WORD EM91
(6) 021126 026606 000000              .WORD ERR13
2342 021130 000461 000000              BR     8$
2343
2344 021132 012701 100000      44$: MOV    #100000,R1    ; POINT TO BUF1 IN EXT MEMORY.
2345 021136 012702 101000              MOV    #101000,R2    ; DITTO BUF2.
2346 021142 005003 000000              CLR    R3             ; WORD NUMBER.
2347 021144 012704 000377              MOV    #255.,R4      ; LOOP COUNT.
2348 021150 005237 177572              INC    MMRO           ; ***** KT ON *****
2349 021154 012137 003432      5$:  MOV    (R1)+,GDDAT   ;
2350 021160 012237 003434              MOV    (R2)+,BDDAT   ;
2351 021164 023737 003432      003434  CMP    GDDAT,BDDAT   ; COMPARE DATA.
2352 021172 001002 000000              BNE    6$            ; BR IF WRONG.
2353 021174 005203 000000              INC    R3             ; BUMP WORD COUNT.
2354 021176 077412 000000              SOB    R4,5$         ; LOOP.
2355 021200 005037 177572      6$:  CLR    MMRO           ; ***** KT OFF *****
2356 021204 005704 000000              TST    R4             ; LOOP COMPLETED (NO ERRORS) ??
2357 021206 001432 000000              BEQ    8$            ; BR IF SO.
2358 021210 010337 003424              MOV    R3,TMPO       ; NO, SAVE FAILING WORD NUMBER.

```


CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 38-2
29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.

SEQ 0091

2359	021214	012703	000006		MOV	#6,R3					
2360	021220	013737	172350	003430	MOV	PAR4,TMP2					
2361	021226	005037	003426		CLR	TMP1					
2362	021232	006337	003430		ASL	TMP2					
2363	021236	006137	003426	7\$:	ROL	TMP1					
2364	021242	077305			SOB	R3,7\$					
2365	021244	042702	160000		BIC	#+C17777,R2					
2366	021250	162702	000002		SUB	#2,R2					
2367	021254	060237	003430		ADD	R2,TMP2					
2368	021260	005537	003426		ADC	TMP1					
2369	021264				DFERR	EM92,ERR14					
(5)	021264	104455							TRAP	C\$ERDF	
(6)	021266	000234							.WORD	156	
(6)	021270	034352							.WORD	EM92	
(6)	021272	026616							.WORD	ERR14	
2370											
2371	021274			8\$:	CKLOOP				TRAP	C\$CLP1	
(3)	021274	104406									
2372	021276				ENDSEG						
(3)	021276										
(3)	021276	104405							10000\$:	TRAP	C\$ESEG
2373	021300	062737	020000	010416	ADD	#20000,BA16					
2374	021306	005537	010420		ADC	BA22					
2375	021312	062737	000200	172350	ADD	#200,PAR4					
2376	021320	023727	172350	9\$:	CMP	PAR4,(PC)+					
2377	021324	007400		IOPAG:	7400						
2378											
2379	021326	103002			BHIS	EXMDUN					
2380	021330	000137	020662		JMP	EXMLUP					
2381											
2382	021334	012737	004366	010416	EXMDUN:	MOV	#BUF1,BA16				
2383	021342	005037	010420		CLR	BA22					
2384	021346				ENDTST						
(3)	021346										
(3)	021346	104401							L10054:	TRAP	C\$ETST

; NOW ASSEMBLE BAD PA.
; ADJUST PAGE TO BIT 21...
; ...STRIP PAR BITS FROM VA...
; ...AND COMBINE THE PIECES.
; DATA ERROR IN EXTENDED MEM.

; INCR PHYSICAL ADDRESS BY 4K...
; ...AND OVERFLOW INTO BAE.
; INCR PAR4 BY 4K ALSO.
; QUIT AT THE I/O PAGE...
; ...007400 (124K)...
; ...OR 177400 (2044K).
; EXIT IF DONE.
; LOOP OTHERWISE.

```

2386 .SBTTL
2387 .SBTTL RLV12 DRIVE INTERFACE TESTS (G5388 TLM REQUIRED).
2388 .SBTTL
2389 021350 STARS
(2) ;*****
2390 ;* ALL THESE TESTS REQUIRE THE SERVICES OF THE G5388 TEST LOOP MODULE.
2391 ;*
2392 .SBTTL 30 -- SYS CLK, PWR OK, DRIVE SELECT, READY, AND ERROR BITS.
2393 ;*
2394 ;* 1. TEST SYS CLOCK AND PWR OK FROM THE RLV12.
2395 ;* 2. TEST DRIVE READY AND DRIVE ERROR TO THE RLV12.
2396 ;* 3. TEST DRIVE SELECT BITS FROM THE RLV12.
2397 021350 STARS
(2) ;*****
2398 021350 TLM1: BEGIN.TEST
(4) 021350 T30::
2399 021350 004737 024034 JSR PC,TLMOK ; CHECK FOR V12 AND TLM OK...
2400 021354 102561 BVS 7$ ; ...AND EXIT IF NOT.
2401 021356 012777 000001 161760 MOV #1,@TCSR ; RESET...
2402 021364 005077 161754 CLR @TCSR ; ...AND CLEAR TLM CSR (LO BYTE).
2403 021370 017737 161750 003434 MOV @TCSR,BDDAT ; GET TLM STATUS.
2404 021376 012737 004400 003432 MOV #4400,GDDAT ; EXPECT SYSCLK<11> AND PWROK<8>.
2405 021404 023737 003432 003434 CMP GDDAT,BDDAT
2406 021412 001404 BEQ 1$ ; BR IF STATUS IS RIGHT.
2407 021414 DFERR EM101,ERR2 ; PWR-OK AND/OR SYS-CLK NOT SET IN TLM.
(5) 021414 104455 TRAP C$ERDF
(6) 021416 000235 .WORD 157
(6) 021420 035061 .WORD EM101
(6) 021422 026272 .WORD ERR2
2408 021424 017700 161674 1$: MOV @RLCS,RO ; GET RLV12 CSR.
2409 021430 010037 003434 MOV RO,BDDAT ; REC'D RLV12 STATUS...
2410 021434 042700 000001 BIC #DRDY,RO ; ...SHOULD HAVE DRDY CLEAR.
2411 021440 010037 003432 MOV RO,GDDAT
2412 021444 023737 003432 003434 CMP GDDAT,BDDAT
2413 021452 001404 BEQ 2$ ; BR IF SO.
2414 021454 DFERR EM102,ERR2 ; DRIVE READY NOT CLEAR IN RLV12.
(5) 021454 104455 TRAP C$ERDF
(6) 021456 000236 .WORD 158
(6) 021460 035142 .WORD EM102
(6) 021462 026272 .WORD ERR2
2415 021464 012777 000200 161652 2$: MOV #200,@TCSR ; SET DRIVE READY IN TLM.
2416 021472 017700 161626 MOV @RLCS,RO ; GET RLV12 AGAIN.
2417 021476 010037 003434 MOV RO,BDDAT
2418 021502 052700 000001 BIS #DRDY,RO ; DRDY SHOULD BE SET NOW.
2419 021506 010037 003432 MOV RO,GDDAT
2420 021512 023737 003432 003434 CMP GDDAT,BDDAT
2421 021520 001404 BEQ 3$ ; BR IF SO.
2422 021522 DFERR EM102,ERR2 ; DRIVE READY NOT SET IN RLV12.
(5) 021522 104455 TRAP C$ERDF
(6) 021524 000237 .WORD 159
(6) 021526 035142 .WORD EM102
(6) 021530 026272 .WORD ERR2
2423 021532 012777 000100 161604 3$: MOV #100,@TCSR ; CLEAR DRDY, SET DERR IN TLM.
2424 021540 017700 161560 MOV @RLCS,RO ; ONE MORE TIME.
2425 021544 010037 003434 MOV RO,BDDAT
2426 021550 052700 140000 BIS #ERR!DERR,RO ; SHOULD HAVE THESE ERROR BITS.

```

CVRLB80 -- RLV12 DISKLESS.
CVRLB8.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 39-1
30 -- SYS CLK, PWR OK, DRIVE SELECT, READY, AND ERROR BITS.

SEQ 0093

```

2427 021554 010037 003432      MOV      R0,GDDAT
2428 021560 023737 003432 003434  CMP      GDDAT,BDDAT
2429 021566 001404      BEQ      4$ ; BR IF SO.
2430 021570      DFERR    EM103,ERR2 ; DRIVE ERROR STATUS INCORRECT IN RLV12.
(5) 021570 104455      TRAP     C$ERDF
(6) 021572 000240      .WORD   160
(6) 021574 035217      .WORD   EM103
(6) 021576 026272      .WORD   ERR2
2431 021600 005077 161540      4$:     CLR      @TCSR ; CLEAR TLM.
2432 021604 005001      CLR      R1 ; INIT DRIVE SELECT BITS<9:8>...
2433 021606 112737 000060 035307  MOVB    #'0,EM104X ;...AND ERROR TEXT.
2434 021614 012703 000004      MOV      #4,R3 ; LOOP CONTROL.
2435 021620 012777 000200 161476  5$:     MOV      #CRDY,@RLCS ; INIT RLV CSR...
2436 021626 050177 161472      BIS      R1,@RLCS ;...INSERT DRIVE BITS <9:8>.
2437 021632 017702 161506      MOV      @TCSR,R2 ; GET TLM CSR.
2438 021636 010237 003434      MOV      R2,BDDAT ; RECV'D TLM SHOULD ECHO DRIVE...
2439 021642 042702 003000      BIC      #BIT10!BIT9,R2 ;...SELECTED IN TLM<10:9>.
2440 021646 006301      ASL      R1
2441 021650 050102      BIS      R1,R2 ; SET EXPECTED VALUE.
2442 021652 006201      ASR      R1
2443 021654 010237 003432      MOV      R2,GDDAT
2444 021660 023737 003432 003434  CMP      GDDAT,BDDAT
2445 021666 001404      BEQ      6$
2446 021670      DFERR    EM104,ERR2 ; DRIVE SELECT FAILS IN TLM.
(5) 021670 104455      TRAP     C$ERDF
(6) 021672 000241      .WORD   161
(6) 021674 035301      .WORD   EM104
(6) 021676 026272      .WORD   ERR2
2447 021700 062701 000400      6$:     ADD      #DS1,R1 ; BUMP DRIVE SELECT NUMBER...
2448 021704 105237 035307      INCB    EM104X ;...AND THE ERROR TEXT.
2449 021710 077335      SOB     R3,5$
2450 021712 012777 000200 161404  MOV      #CRDY,@RLCS ; RESET DRIVE NUMBER IN RL.
2451 021720      7$:     ENDTST
(3) 021720      L10055:
(3) 021720 104401      TRAP     C$ETST

```



```

2453          .SBTTL 31 -- DRIVE COMMAND, STATUS AND STATUS CLOCK.
2454
2455 021722    STARS
(2)          ;:*****
2456          ;* NOW TEST DRIVE COMMAND, STATUS, AND STATUS CLOCK
2457          ;* USING A "GET STATUS" COMMAND.
2458 021722    STARS
(2)          ;:*****
2459 021722    TLM2: BEGIN.TEST
(4) 021722          T31::
2460 021722    004737 024034      JSR    PC,TLMOK
2461 021726    102476          BVS    5$          ; EXIT IF NO GOOD.
2462 021730    004737 024204      JSR    PC,CLRTLM   ; RESET TLM, AND SET DRDY.
2463 021734    005037 003436      CLR    INIMP
2464 021740    012765 125252 000004  MOV    #125252,4(R5) ; SET PHONEY STATUS WORD.
2465 021746    012777 000204 161350  MOV    #CRDY!GSTAT,@RLCS ; SET "GET STATUS" COMMAND...
2466 021754    012777 000013 161346  MOV    #DRST!GSBIT!MK,@RLDA ; ...AND RESET STATUS BITS.
2467 021762    004537 010062          JSR    R5,BEFORE
2468 021766    042777 000200 161330  BIC    #CRDY,@RLCS ; XCT IT.
2469 021774    005000          CLR    RO
2470 021776    011501          1$: MOV    (R5),R1      ; GET TCSR...
2471 022000    100401          BMI    2$          ; ...PROCEED WHEN "NEW SKGS" SETS...
2472 022002    077003          SOB    RO,1$      ; ...BUT DON'T WAIT FOREVER !!!
2473 022004    016537 000002 003434  2$: MOV    2(R5),BDDAT ; SAVE RECV'D DRIVE COMMAND (SKGS)...
2474 022012    004537 010704          JSR    R5,WTCRDY
2475 022016    012737 000013 003432  MOV    #DRST!GSBIT!MK,GDDAT ; ...SHOULD LOOK LIKE THIS.
2476 022024    023737 003432 003434  CMP    GDDAT,BDDAT
2477 022032    001404          BEQ    3$          ; BR IF STATUS COMMAND WAS RIGHT.
2478 022034          DFERR  EM105,ERR4 ; GET STATUS COMMAND WRONG IN TLM.
(5) 022034    104455          TRAP   C$ERDF
(6) 022036    000242          .WORD 162
(6) 022040    035363          .WORD EM105
(6) 022042    026356          .WORD ERR4
2479 022044    013737 003372 003434  3$: MOV    E.MP,BDDAT ; RETURNED DRIVE STATUS...
2480 022052    012737 125252 003432  MOV    #125252,GDDAT ; ...SHOULD = PHONEY.
2481 022060    023737 003432 003434  CMP    GDDAT,BDDAT
2482 022066    001404          BEQ    4$          ; BR IF RETURNED STATUS IS RIGHT.
2483 022070          DFERR  EM106,ERR4 ; RETURNED STATUS INCORRECT IN RLMP.
(5) 022070    104455          TRAP   C$ERDF
(6) 022072    000243          .WORD 163
(6) 022074    035445          .WORD EM106
(6) 022076    026356          .WORD ERR4
2484 022100    004537 007554          4$: JSR    R5,GETERR ; FINAL CONTROLLER STATUS...
2485 022104    000407          BR     5$          ; ...SHOULD BE ERROR FREE.
2486 022106    012737 036742 003426  MOV    #EXPNON,TMP1
2487 022114          DFERR  EM107,ERR20A ; RLV12 ERRORS AFTER RESET GET STATUS
(5) 022114    104455          TRAP   C$ERDF
(6) 022116    000244          .WORD 164
(6) 022120    035515          .WORD EM107
(6) 022122    026740          .WORD ERR20A
2488 022124          5$: ENDTST
(3) 022124          L10056: TRAP   C$ETST
(3) 022124    104401

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 41
32 -- DRIVE COMMAND, SEEK DIFF AND SECTOR PULSE.

SEQ 0095

.SBTTL 32 -- DRIVE COMMAND, SEEK DIFF AND SECTOR PULSE.

2490
2491
2492 022126
(2)
2493
2494 022126
(2)
2495 022126
(4) 022126
2496 022126 004737 024034
2497 022132 102457
2498 022134 004737 024204
2499 022140 005037 003436
2500 022144 052715 000024
2501 022150 012777 000206 161146
2502 022156 012777 077601 161144
2503 022164 004537 010062
2504 022170 042777 000200 161126
2505 022176 005000
2506 022200 011501 1\$:
2507 022202 100401
2508 022204 077003
2509 022206 016537 000002 003434 2\$:
2510 022214 004537 010704
2511 022220 012737 077601 003432
2512 022226 023737 003432 003434
2513 022234 001404
2514 022236
(5) 022236 104455
(6) 022240 000245
(6) 022242 035570
(6) 022244 026356
2515 022246 004537 007554 3\$:
2516 022252 000407
2517 022254 012737 036742 003426
2518 022262
(5) 022262 104455
(6) 022264 000246
(6) 022266 035651
(6) 022270 026740
2519 022272 4\$:
(3) 022272
(3) 022272 104401

```

STARS
;*****
;* TEST SECTOR PULSE USING A "SEEK" COMMAND.
STARS
;*****
TLM3: BEGIN.TEST
T32::
JSR PC,TLMOK
BVS 4$ ; BR IF NOT OK.
JSR PC,CLRTLM ; RESET TTLM, SET DRDY.
CLR INIMP
BIS #24,(R5) ; ENABLE SECTOR GENERATOR (NOM CLOCK).
MOV #CRDY!SEEK,@RLCS ; SET "SEEK" COMMAND...
MOV #77600!MK,@RLDA ; ...AND A PHONEY SEEK DIFFERENCE.
JSR R5,BEFORE
BIC #CRDY,@RLCS ; XCT IT.
CLR R0
MOV (R5),R1 1$:
BMI 2$ ; PROCEED WHEN "NEW SKGS" SETS...
SOB R0,1$ ; ...KEEP-ALIVE !!
MOV 2(R5),BDDAT ; SAVE RECV'D DRIVE COMMAND...
JSR R5,WTCRDY
MOV #77600!MK,GDDAT ; ...SHOULD = PHONEY DIFFERENCE.
CMP GDDAT,BDDAT
BEQ 3$ ; BR IF SO.
DFERR EM110,ERR4 ; DRIVE COMMAND WRONG IN TLM.
TRAP C$ERDF
.WORD 165
.WORD EM110
.WORD ERR4
JSR R5,GETERR ; FINAL CONTROLLER STATUS...
BR 4$ ; ...SHOULD BE ERROR FREE.
MOV #EXPNON,TMP1
DFERR EM111,ERR20A ; RLV ERRORS AFTER SEEK COMMAND
TRAP C$ERDF
.WORD 166
.WORD EM111
.WORD ERR20A
4$: ENDTST
L10057: TRAP C$ETST

```

```

2521          .SBTTL 33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.
2522
2523 022274    STARS
(2)          ;:*****
2524          ;* TEST THE WRITE GATE, WRITE DATA ACTIVE, AND WRITE GATE ERROR.
2525 022274    STARS
(2)          ;:*****
2526 022274    TLM4: BEGIN.TEST
(4) 022274    T33::
2527 022274 004737 024034 JSR PC,TLMOK
2528 022300 102524 BVS 5$ ; BR IF NOT OK.
2529 022302 004737 024204 JSR PC,CLRTLM ; RESET TLM, SET DRDY.
2530 022306 052715 000024 BIS #24,(R5) ; ENABLE SECTOR GEN (NOM CLOCK).
2531 022312 004537 010224 JSR R5,WDELAY ; DELAY SO THAT WE ENTER THE TLM...
2532 022316 000002 2 ; ...SEQUENCE ASYNCHRONOUSLY.
2533 022320 012777 000212 160776 MOV #CRDY!WRITE,@RLCS ; SET COMMAND TO WRITE...
2534 022326 012737 177600 003436 MOV #-128.,INIMP
2535 022334 013777 003436 160770 MOV INIMP,@RLMP ; ...128 WORDS...
2536 022342 012777 004366 160756 MOV #BUF1,@RLBA ; ...FROM BUF1...
2537 022350 012777 000205 160752 MOV #205,@RLDA ; ...TO CYL 1, SECT 5.
2538 022356 004537 010062 JSR R5,BEFORE
2539 022362 042777 000200 160734 BIC #CRDY,@RLCS ; XCT IT.
2540 022370 005000 CLR R0
2541 022372 011501 1$: MOV (R5),R1 ; TLM STATUS => R1.
2542 022374 010137 003434 MOV R1,BDDAT ; RECV'D TLM STATUS...
2543 022400 052701 030000 BIS #BIT13!BIT12,R1 ; ...WILL GET WRITE GATE AND DATA ACTIVE.
2544 022404 010137 003432 MOV R1,GDDAT
2545 022410 023737 003432 003434 CMP GDDAT,BDDAT
2546 022416 001410 BEQ 2$ ; BR IF AND WHEN THAT OCCURS...
2547 022420 077014 SOB R0,1$ ; ...WITHIN REASON -- OF COURSE.
2548 022422 004537 010704 JSR R5,WTCRDY
2549 022426 004537 010704 DFERR EM112,ERR4 ; WRITE GATE AND/OR DATA ACTIVE NOT SET.
(5) 022426 104455 TRAP C$ERDF
(6) 022430 000247 .WORD 167
(6) 022432 035716 .WORD EM112
(6) 022434 026356 .WORD ERR4
2550 022436 000402 SKP2
2551 022440 004537 010704 2$: JSR R5,WTCRDY
2552 022444 011537 003434 MOV (R5),BDDAT ; FINAL TLM STATUS...
2553 022450 013737 003434 003432 MOV BDDAT,GDDAT
2554 022456 042737 060000 003432 BIC #BIT14!BIT13,GDDAT ; ...ERROR AND GATE SHOULD BE CLEAR.
2555 022464 023737 003432 003434 CMP GDDAT,BDDAT
2556 022472 001415 BEQ 4$ ; BR IF SO.
2557 022474 032737 020000 003434 BIT #BIT13,BDDAT ; IT'S NOT, IS GATE STILL SET ??
2558 022502 001405 BEQ 3$ ;
2559 022504 DFERR EM113A,ERR4 ; WRITE GATE STILL SET AFTER DONE.
(5) 022504 104455 TRAP C$ERDF
(6) 022506 000250 .WORD 168
(6) 022510 036062 .WORD EM113A
(6) 022512 026356 .WORD ERR4
2560 022514 000404 BR 4$
2561 022516 3$: DFERR EM113,ERR4 ; WRITE GATE ERROR IN TLM.
(5) 022516 104455 TRAP C$ERDF
(6) 022520 000251 .WORD 169
(6) 022522 036010 .WORD EM113
(6) 022524 026356 .WORD ERR4

```


CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 42-1
33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.

SEQ 0097

2562	022526	004537	007554	4\$:	JSR	R5,GETERR	; FINAL RLV STATUS...		
2563	022532	000407			BR	5\$;...SHOULD BE ERROR FREE.		
2564	022534	012737	036742	003426	MOV	#EXPNON,TMP1			
2565	022542				DFERR	EM114,ERR20A	; RL ERROR AFTER WRITE.		
(5)	022542	104455						TRAP	C\$ERDF
(6)	022544	000252						.WORD	170
(6)	022546	036140						.WORD	EM114
(6)	022550	026740						.WORD	ERR20A
2566	022552			5\$:	ENDTST				
(3)	022552							L10060:	
(3)	022552	104401						TRAP	C\$ETST

2568
2569
2570 022554
(2)
2571
2572
2573
2574
2575
2576 022554

.SBTTL 34 -- READ DATA, READ HEADER, AND READ DATA W/O HEADER.

STARS

;* READ AND VERIFY DATA ON EACH OF 6 TLM SECTORS.
;* SECTOR 3 GETS A READ HEADER AND READ DATA W/O HEADER SEQUENCE.
;* REPEAT USING ALTERNATE VCO SETTINGS (WITH MAX AND MIN PEAK SHIFT).
;* VARIOUS PEAK SHIFT, CLOCK, AND SECTOR OPTIONS MAY BE SELECTED VIA
;* THE "CHANGE SOFTWARE" DIALOGUE AT START/RESTART TIME.

STARS

TLM5: BEGIN.TEST

2577 022554
(4) 022554
2578 022554 004737 024040
2579 022560 102525
2580 022562 012737 177600 003436
2581 022570 013737 003314 022616
2582 022576 042737 177770 022616
2583 022604 001002
2584 022606 005237 022616
2585 022612 004737 024266
2586 022616 000001
2587 022620 102010
2588 022622 113737 036604 036573
2589 022630

T34::
; VERIFY TLM OK AND PROM ID.
; EXIT IF NOT OK.
; INIT WORD COUNT.
; GET SECTOR NUMBER (IF ANY).
; NZ = SINGLE SECTOR MODE.
; OTHERWISE, USE ALL STARTING WITH 1.
; SET BUFFERS AND DISK ADDRESS...
; ...FOR THIS SECTOR NUMBER.
; BR IF SECTOR FOUND.
; SET ASCII SECTOR NUMBER.
; CAN'T FIND SECTOR.

(5) 022630 104455
(6) 022632 000253
(6) 022634 036526
(6) 022636 026232
2590 022640 000465
2591 022642 032737 000004 003312 1\$:
2592 022650 001006
2593 022652 012715 000224
2594 022656 012737 036610 003430
2595 022664 000423
2596 022666 032737 000002 003312 2\$:
2597 022674 001006
2598 022676 012715 000230
2599 022702 012737 036626 003430
2600 022710 000411
2601 022712 032737 000001 003312 3\$:
2602 022720 001035
2603 022722 012715 000220
2604 022726 012737 036641 003430

DFERR EM119,NOSIG
; TRAP C\$ERDF
; .WORD 171
; .WORD EM119
; .WORD NOSIG
BR 7\$
; NOMINAL CLOCK ENABLED ??
; BR IF NOT.
; YES, SET NOMINAL CLOCK, NO PEAK SHIFT.
BIT #BIT2,MPXCLK 2\$
MOV #224,(R5)
; FAST CLOCK ENABLED ??
; BR IF NOT.
; YES, SET FAST CLOCK, NO PEAK SHIFT.
#TCLK0,TMP2 4\$
BIT #BIT1,MPXCLK 3\$
; SLOW CLOCK ENABLED ??
; BR IF NOT.
; YES, SET SLOW CLOCK, NO PEAK SHIFT.
#230,(R5)
#TCLK1,TMP2 4\$
BR #BIT0,MPXCLK 7\$
; MIN PEAK SHIFT ENABLED ??
; BR IF NOT.
; YES, PEAK SHIFT IS OFF...
; ...READ SECTOR AND VERIFY DATA.
; MAX PEAK SHIFT ENABLED ??
; BR IF NOT.
; YES, MAX PEAK SHIFT ON...
; ...EXTEND CLOCK MESSAGE...
; ...DO IT ALL AGAIN.
#220,(R5)
#TCLK2,TMP2

2605
2606 022734 032737 040000 003312 4\$:
2607 022742 001004
2608 022744 105037 036654
2609 022750 004737 023040
2610 022754 005737 003312 5\$:
2611 022760 100407
2612 022762 052715 000040
2613 022766 112737 000040 036654
2614 022774 004737 023040
2615
2616 023000 032715 000004 6\$:

MOV #40,(R5)
; NOMINAL DONE ??
#4,(R5)
; NOMINAL DONE ??
PC,READSEC
MPXCLK 6\$
BIS #40,(R5)
; ...EXTEND CLOCK MESSAGE...
; ...DO IT ALL AGAIN.
PC,READSEC
TST MPXCLK 5\$
CLR B
TPEAK
; YES, PEAK SHIFT IS OFF...
; ...READ SECTOR AND VERIFY DATA.
; MAX PEAK SHIFT ENABLED ??
; BR IF NOT.
; YES, MAX PEAK SHIFT ON...
; ...EXTEND CLOCK MESSAGE...
; ...DO IT ALL AGAIN.
; SLOW CLOCK ENABLED ??
; BR IF NOT.
; YES, SET SLOW CLOCK, NO PEAK SHIFT.
; FAST CLOCK ENABLED ??
; BR IF NOT.
; YES, SET FAST CLOCK, NO PEAK SHIFT.
; NOMINAL CLOCK ENABLED ??
; BR IF NOT.
; YES, SET NOMINAL CLOCK, NO PEAK SHIFT.

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33MACY11 30(1046) 12-JUN-85 15:42 PAGE 43-1
34 -- READ DATA, READ HEADER, AND READ DATA W/O HEADER.

SEQ 0099

```

2617 023004 001330          BNE      2$          ; LOOP TO "FAST" IF SO.
2618 023006 032715 000010  BIT      #10,(R5)    ; FAST DONE ??
2619 023012 001337          BNE      3$          ; LOOP TO "SLOW" IF SO...
2620                                ; ...OTHERWISE, THIS SECTOR IS DONE.
2621
2622 023014 032737 000007 003314 7$:   BIT      #7,SNGLSEC  ; SINGLE SECTOR MODE ??
2623 023022 001004          BNE      8$          ; WE'RE DONE IF SO.
2624 023024 023727 022616 000006  CMP      12$,#6      ; NO, LAST SECTOR DONE ??
2625 023032 103665          BLO      10$         ; GO "ROUND AGAIN IF NOT...
2626 023034          8$:   EXIT     TST          ; ...OTHERWISE, NEXT TEST.
(3) 023034 104432                                TRAP    C$EXIT
(3) 023036 000376                                .WORD  L10061-.
2627
2628 ; SUBROUTINE TO READ DATA FROM TLM TO BUF2 AND VERIFY (AGAINST BUF1).
2629 ; DISK ADDRESS AND EXPECTED DATA (BUF1) MUST BE PRESET BEFORE CALL.
2630 ; WORD COUNT (-128.) IS PRESET FOR ALL.
2631
2632 023040          ; READSEC: BGNSEG                                TRAP    C$BSEG
(3) 023040 104404
2633 023042 012700 005366          MOV      #BUF2,R0
2634 023046 012701 000200          MOV      #128.,R1
2635 023052 012720 177777 1$:   MOV      #-1,(R0)+  ; CLEAR RECEIVING BUFFER.
2636 023056 077103          SOB      R1,1$
2637 023060 105037 036257          CLRB    EM115X      ; INIT ERROR TEXT FOR "READ".
2638 023064 105037 036427          CLRB    EM117X
2639 023070 012777 000214 160226  MOV      #CRDY!READ,@RLCS ; SET COMMAND TO READ...
2640 023076 013777 003436 160226  MOV      INIMP,@RLMP  ; ...128 WORDS...
2641 023104 013777 003440 160216  MOV      INIDA,@RLDA  ; ...FROM CYL X, SECT Y...
2642 023112 012777 005366 160206  MOV      #BUF2,@RLBA  ; ...TO BUF2.
2643 023120 004537 010062 2$:   JSR      R5,BEFORE
2644 023124 042777 000200 160172  BIC      #CRDY,@RLCS ; XCT.
2645 023132 004537 010704          JSR      R5,WTCRDY   ; WAIT FOR RL DONE.
2646 023136 123737 024560 003365  CMPB    PSERR,E.CS+1 ; FINAL STATUS RIGHT ??
2647 023144 001412          BEQ      3$          ; BR IF SO.
2648 023146 004537 007554          JSR      R5,GETERR   ; NO, GET WHATEVER'S THERE.
2649 023152 000240          NOP
2650 023154 013737 024556 003426  MOV      PSETXT,TMP1
2651 023162          DFERR   EM115,ERR20  ; ERROR STATUS WRONG ON READ.
(5) 023162 104455                                TRAP    C$ERDF
(6) 023164 000254                                .WORD  172
(6) 023166 036211                                .WORD  EM115
(6) 023170 026734                                .WORD  ERR20
2652 023172 023727 024556 036676 3$:   CMP      PSETXT,#EXPHCRC ; NOW IS THIS THE BAD HEADER SECTOR ??
2653 023200 001062          BNE      CHKDATA    ; BR IF NOT.
2654
2655 023202 012700 000007          READNH: MOV      #7,R0      ; YES IT IS, SET UP TO READ HEADERS.
2656 023206 112737 000040 036257  MOVB    #40,EM115X  ; EXTEND ERROR TEXT TO "READ NH"
2657 023214 112737 000040 036427  MOVB    #40,EM117X
2658 023222 012777 000210 160074 1$:   MOV      #CRDY!RDHDR,@RLCS
2659 023230 042777 000200 160066  BIC      #CRDY,@RLCS ; XCT READ HEADER...
2660 023236 105777 160062 2$:   TSTB    @RLCS      ; ...AND WAIT FOR DONE.
2661 023242 100375          BPL      2$
2662 023244 027727 160062 000206  CMP      @RLMP,#<1_7.>!6 ; FOUND BAD SECTOR-1 ??
2663 023252 001406          BEQ      3$          ; BR IF SO.
2664 023254 077016          SOB      R0,1$
2665 023256          DFERR   EM116,ERR21  ; CAN'T FIND CYL 1, SECT 6

```



```

(5) 023256 104455 TRAP C$ERDF
(6) 023260 000255 .WORD 173
(6) 023262 036305 .WORD EM116
(6) 023264 027014 .WORD ERR21
2666 023266 000460 BR READDUN ; ABORT THE REST.
2667 023270 012777 000216 160026 3$: MOV #CRDY!RDNHDR,@RLCS ; OK, SET READ-NO-HEADER COMMAND...
2668 023276 013777 003436 160026 MOV INIMP,@RLMP ; ...INSURE WORD COUNT IS STILL THERE...
2669 023304 004537 010062 JSR R5,BEFORE ; ...AND XCT IT.
2670 023310 042777 000200 160006 BIC #CRDY,@RLCS ; ...AND XCT IT.
2671 023316 004537 010704 JSR R5,WTCRDY
2672 023322 004537 007554 JSR R5,GETERR ; SHOULD HAVE NO ERRORS...
2673 023326 000407 BR CHKDATA ; ...BR IF SO.
2674 023330 012737 036742 003426 MOV #EXPNON,TMP1
2675 023336 DFERR EM115,ERR20 ; ERROR STATUS INCORRECT AFTER READ NH
(5) 023336 104455 TRAP C$ERDF
(6) 023340 000256 .WORD 174
(6) 023342 036211 .WORD EM115
(6) 023344 026734 .WORD ERR20
2676
2677 023346 012700 004446 CHKDATA: MOV #BUF1+48.,R0 ; EXPD DATA FIELD BEGINS AT WORD 24...
2678 023352 012701 005366 MOV #BUF2,R1 ; ...RECD DATA AT WORD 0.
2679 023356 005002 CLR R2 ; ...AND WORD NUMBER.
2680 023360 021011 1$: CMP (R0),(R1) ; COMPARE EXP'D VS REC'D DATA...
2681 023362 001415 BEQ 2$ ; ...AND BR IF OK.
2682 023364 011037 003432 MOV (R0),GDDAT ; IT'S NOT, SET EXP'D...
2683 023370 011137 003434 MOV (R1),BDDAT ; ...RECV'D...
2684 023374 010137 003424 MOV R1,TMP0 ; ...ADDRESS...
2685 023400 010237 003426 MOV R2,TMP1 ; ...AND WORD NUMBER.
2686 023404 DFERR EM117,ERR22 ; DATA INCORRECT ON READ OR READ NH.
(5) 023404 104455 TRAP C$ERDF
(6) 023406 000257 .WORD 175
(6) 023410 036365 .WORD EM117
(6) 023412 027024 .WORD ERR22
2687 023414 000405 BR 3$
2688 023416 022021 2$: CMP (R0)+,(R1)+ ; BUMP ADDRESSES...
2689 023420 005202 INC R2 ; ...AND WORD NUMBER.
2690 023422 032702 000177 BIT #177,R2
2691 023426 001354 BNE 1$ ; ...AND LOOP 'TIL DONE.
2692 023430 3$: READDUN: ENDSEG
2693 023430
(3) 023430 10000$: TRAP C$ESEG
(3) 023430 104405
2694 023432 000207 RTS PC
2695 023434 ENDTST
(3) 023434 L10061: TRAP C$ETST
(3) 023434 104401

```

```
.SBTTL 35 -- WRITE CHECK.

2697
2698
2699 023436 STARS
(2) ;*****
2700 ;* DO A WRITE CHECK USING SECTOR AND CLOCK OPTIONS AS BEFORE.
2701 023436 STARS
(2) ;*****
2702 023436 TLM6: BEGIN.TEST
(4) 023436 T35::
2703 023436 004737 024040 JSR PC,TLMPID
2704 023442 102525 BVS 8$ ; EXIT IF NOT OK.
2705 023444 012737 177600 003436 MOV #-128.,INIMP ; INIT WORD COUNT.
2706 023452 013737 003314 023500 MOV SNGLSEC,12$ ; GET SECTOR NUMBER (IF ANY).
2707 023460 042737 177770 023500 BIC #7,12$
2708 023466 001002 BNE 11$ ; NZ = SINGLE SECTOR MODE.
2709 023470 005237 023500 10$: INC 12$ ; OTHERWISE, USE ALL STARTING WITH 1.
2710 023474 004737 024266 11$: JSR PC,PSX ; SET BUFFERS AND DISK ADDRESS...
2711 023500 000001 12$: 1 ;...FOR THIS SECTOR NUMBER.
2712 023502 102010 BVC 1$
2713 023504 113737 036604 036573 MOVB TSECX,EM119X ; SET ASCII SECTOR NUMBER.
2714 023512 DFERR EM119,NOSIG ; CAN'T FIND SECTOR.
(5) 023512 104455 TRAP C$ERDF
(6) 023514 000260 .WORD 176
(6) 023516 036526 .WORD EM119
(6) 023520 026232 .WORD NOSIG
2715 023522 000465 BR 7$
2716 023524 032737 000004 003312 1$: BIT #BIT2,MPXCLK ; NOMINAL CLOCK ENABLED ??
2717 023532 001006 BNE 2$ ; BR IF NOT.
2718 023534 012715 000224 MOV #224,(R5) ; YES, SET NOMINAL CLOCK, NO PEAK SHIFT.
2719 023540 012737 036610 003430 MOV #TCLK0,TMP2
2720 023546 000423 BR 4$
2721 023550 032737 000002 003312 2$: BIT #BIT1,MPXCLK ; FAST CLOCK ENABLED ??
2722 023556 001006 BNE 3$ ; BR IF NOT.
2723 023560 012715 000230 MOV #230,(R5) ; YES, SET FAST CLOCK, NO PEAK SHIFT.
2724 023564 012737 036626 003430 MOV #TCLK1,TMP2
2725 023572 000411 BR 4$
2726 023574 032737 000001 003312 3$: BIT #BIT0,MPXCLK ; SLOW CLOCK ENABLED ??
2727 023602 001035 BNE 7$ ; BR IF NOT.
2728 023604 012715 000220 MOV #220,(R5) ; YES, SET SLOW CLOCK, NO PEAK SHIFT.
2729 023610 012737 036641 003430 MOV #TCLK2,TMP2
2730
2731 023616 032737 040000 003312 4$: BIT #BIT14,MPXCLK ; MIN PEAK ENABLED ??
2732 023624 001004 BNE 5$ ; BR IF NOT.
2733 023626 105037 036654 CLR B TPEAK ; YES, PEAK SHIFT IS OFF...
2734 023632 004737 023722 JSR PC,WRITCHK ;...WRITE-CHECK THIS SECTOR.
2735 023636 005737 003312 5$: TST MPXCLK ; MAX PEAK SHIFT ENABLED ??
2736 023642 100407 BMI 6$ ; BR IF NOT.
2737 023644 052715 000040 BIS #40,(R5) ; YES, MAX PEAK SHIFT ON...
2738 023650 112737 000040 036654 MOVB #40,TPEAK ;...EXTEND CLOCK MESSAGE...
2739 023656 004737 023722 JSR PC,WRITCHK ;...AND DO IT AGAIN.
2740
2741 023662 032715 000004 6$: BIT #4,(R5) ; NOMINAL DONE ??
2742 023666 001330 BNE 2$ ; LOOP TO "FAST" IF SO.
2743 023670 032715 000010 BIT #10,(R5) ; FAST DONE ??
2744 023674 001337 BNE 3$ ; LOOP TO "SLOW" IF SO...
2745 ;...OTHERWISE, THIS SECTOR IS DONE.
```

```

2746
2747 023676 032737 000007 003314 7$: BIT #7,SNGLSEC ; SINGLE SECTOR MODE ??
2748 023704 001004 BNE 8$ ; WE'RE ALL DONE IF SO.
2749 023706 023727 023500 000006 CMP 12$,#6 ; NO, LAST SECTOR DONE ??
2750 023714 103665 BLO 10$ ; GO "ROUND AGAIN IF NOT...
2751 023716 8$: EXIT TST ; ...OTHERWISE, WE'RE ALL DONE.
(3) 023716 104432 TRAP C$EXIT
(3) 023720 000112 .WORD L10062-.
2752 ;
2753 ; SUBROUTINE TO WRITE CHECK A SECTOR AGAINST THE DATA IN "BUF1".
2754 ;
2755 WRITCHK: BGNSEG TRAP C$BSEG
(3) 023722 104404
2756 023724 012777 000202 157372 MOV #CRDY!WRCHK,@RLCS ; SET COMMAND TO WRITE-CHECK...
2757 023732 013777 003436 157372 MOV INIMP,@RLMP ; ...128 WORDS...
2758 023740 013777 003440 157362 MOV INIDA,@RLDA ; ...FROM CYL X, SECT Y...
2759 023746 012777 004446 157352 MOV #BUF1+48.,@RLBA ; ...AGAINST BUF1.
2760 023754 004537 010062 JSR R5,BEFORE
2761 023760 042777 000200 157336 BIC #CRDY,@RLCS ; XCT.
2762 023766 004537 010704 JSR R5,WTCRDY ; WAIT FOR RL DONE.
2763 023772 123737 024560 003365 CMPB PSERR,E.CS+1 ; FINAL STATUS RIGHT ??
2764 024000 001412 BEQ 1$ ; BR IF SO.
2765 024002 004537 007554 JSR R5,GETERR ; NO, GET WHATEVER'S THERE.
2766 024006 000240 NOP
2767 024010 013737 024556 003426 MOV PSETXT,TMP1
2768 024016 024556 003426 DFERR EM118,ERR20 ; ERROR STATUS WRONG ON WRITE-CHECK
(5) 024016 104455 TRAP C$ERDF
(6) 024020 000261 .WORD 177
(6) 024022 036455 .WORD EM118
(6) 024024 026734 .WORD ERR20
2769 024026 1$: ENDSEG
(3) 024026 10000$: TRAP C$ESEG
(3) 024026 104405
2770 024030 000207 RTS PC
2771 024032 ENDTST
(3) 024032 L10062: TRAP C$ETST
(3) 024032 104401

```



```

2773 ;
2774 ; SUBROUTINE TO SEE IF TLM IS THERE AND PROPERLY CABLED TO RLV12.
2775 ; ALSO CHECK THAT CORRECT PROM IS INSTALLED.
2776 ; REPORT ERROR AND RETURN WITH "V" SET IF NOT.
2777 ;
2778 024034 005000 TLMOK: CLR R0 ; DISABLE ID CHECK.
2779 024036 000401 SKP1
2780 024040 005200 TLMPID: INC R0 ; ENABLE ID CHECK.
2781 024042 013737 003344 003432 MOV TCSR,GDDAT ; GET CSR POINTER.
2782 024050 001453 BEQ 4$ ; ABORT IF IT'S NOT DEFINED.
2783 024052 005037 010672 CLR TRPFLG
2784 024056 005077 157262 CLR @TCSR ; TLM ARE YOU REALLY THERE ??
2785 024062 000240 240
2786 024064 005737 010672 TST TRPFLG
2787 024070 001025 BNE 1$ ; ERROR IF NOT.
2788 024072 012737 004400 003432 MOV #4400,GDDAT ; OTHERWISE, ...
2789 024100 017737 157240 003434 MOV @TCSR,BDDAT ; ...GET INITIAL STATUS.
2790 024106 001423 BEQ 2$ ; IF ZERO, CABLE ISN'T CONNECTED.
2791 024110 005700 TST R0 ; ID VERIFICATION REQUIRED ??
2792 024112 001413 BEQ 5$ ; WE'RE DONE IF NOT.
2793 024114 004737 024204 JSR PC,CLRTLTM ; YES, RESET, ID => R0.
2794 024120 010037 003434 MOV R0,BDDAT
2795 024124 013737 003310 003432 MOV PROMID,GDDAT ; GET EXPECTED ID.
2796 024132 023737 003432 003434 CMP GDDAT,BDDAT
2797 024140 001013 BNE 3$ ; BR IF ID INCORRECT.
2798 024142 000207 5$: RTS PC ; RETURN, "V" = 0.
2799
2800 024144 1$: DFERR EM100,ERR1 ; TIME-OUT ON TLM ADDRESS.
(5) 024144 104455 TRAP C$ERDF
(6) 024146 000262 .WORD 178
(6) 024150 034646 .WORD EM100
(6) 024152 026242 .WORD ERR1
2801 024154 000411 BR 4$
2802 024156 2$: DFERR EM100A,ERR2 ; CABLE PROBABLY NOT INSTALLED.
(5) 024156 104455 TRAP C$ERDF
(6) 024160 000263 .WORD 179
(6) 024162 034702 .WORD EM100A
(6) 024164 026272 .WORD ERR2
2803 024166 000404 BR 4$
2804 024170 3$: DFERR EM100B,ERR2 ; PROM ID INCORRECT.
(5) 024170 104455 TRAP C$ERDF
(6) 024172 000264 .WORD 180
(6) 024174 034765 .WORD EM100B
(6) 024176 026272 .WORD ERR2
2805 024200 000262 4$: SEV ; SET "ERROR" FLAG...
2806 024202 000207 RTS PC ; ...AND RETURN.
2807 ;
2808 ; SUBROUTINE TO RESET TEST LOOP MODULE.
2809 ; RETURN WITH "DRIVE READY" SET, TLM PROM ID IN R0,
2810 ; AND TLM CSR ADDRESS IN R5.
2811 ; AS LONG AS WE'RE HERE, CLEAR RLBA, RLMP, AND RLBAE AS WELL.
2812 ;
2813 024204 013705 003344 CLRTLTM: MOV TCSR,R5 ; DEDICATE R5 TO THE TLM.
2814 024210 012715 000001 MOV #1,(R5) ; RESET...
2815 024214 016500 000006 MOV 6(R5),R0 ; ...PROM ID => R0.
2816 024220 005065 000002 CLR 2(R5) ; CLEAR SKGS...

```

```

2817 024224 005065 000004          CLR      4(R5)          ;...AND PSUEDO-STATUS REGISTERS.
2818 024230 012715 000002          MOV      #2,(R5)       ; CLEAR FLAGS ??
2819 024234 012715 000200          MOV      #200,(R5)    ; SET DRIVE READY.
2820 024240 005077 157062          CLR      @RLBA
2821 024244 005077 157062          CLR      @RLMP
2822 024250 023727 003402 000003  CMP      RLTP,#RLV12X
2823 024256 001002                   BNE      1$
2824 024260 005077 157050          CLR      @RLBAE
2825 024264 000207                   1$:      RTS      PC
2826
2827 ; SUBROUTINE TO SET UP BUFFERS AND DISK ADDRESSES
2828 ; FOR READING EACH OF THE 6 PSUEDO SECTORS.
2829 ; USE HEADER TABLE AND SEARCH ROUTINE IN THE DUMP UTILITY.
2830 ; IF SUCCESSFUL, CONTENTS OF SELECTED SECTOR (160. WORDS) ARE IN "BUF1".
2831 ; CALL: JSR PC,PSX
2832 ; N ; SECTOR NUMBER.
2833 ; BVC XX ; ON RETURN, "V" IS SET IN SECTOR NOT FOUND.
2834 ; ERROR
2835 ; XX: ; CONTINUE.
2836
2837 024266 017600 000000          PSX:    MOV      @(SP),RO      ; GET SECTOR NUMBER (1 - 6)...
2838 024272 005300                   DEC      RO              ;...MAKE IT (0 - 5)...
2839 024274 006300                   ASL      RO              ;...SHIFT UP TO AN INDEX.
2840 024276 062716 000002          ADD      #2,(SP)        ; ADJUST RETURN PC...
2841 024302 000170 024306          JMP      @1$(RO)        ;...AND DO ONE OF THE FOLLOWING:
2842 024306 024324                   1$:      PS1
2843 024310 024342                   PS2
2844 024312 024374                   PS3
2845 024314 024426                   PS4
2846 024316 024444                   PS5
2847 024320 024514                   PS6
2848 024322 024462                   PS7
2849 ; SOONER OR LATER SOME TURKEY WILL...
2850 ;...TRY TO ACCESS SECTOR 7.
2851
2852 024324 112737 000061 036604  PS1:    .ENABL  LSB
2853 024332 012737 000205 003440  MOV      #'1,TSECX      ; TLM SECTOR 1...
2854 024340 000473                   MOV      #<1_7.>!5,INIDA ;...IS CYL 1 SECT 5...
2855 ;...AND IS ERROR FREE.
2856 024342 112737 000062 036604  PS2:    MOV      #'2,TSECX      ; TLM SECTOR 2...
2857 024350 012737 000206 003440  MOV      #<1_7.>!6,INIDA ;...IS CYL 1, SECT 6...
2858 024356 012737 000210 024560  MOV      #H<ERR!DCRC>,PSERR
2859 024364 012737 036722 024556  MOV      #EXPDCRC,PSETXT ;...AND HAS A BAD DATA CRC.
2860 024372 000463                   BR      2$
2861
2862 024374 112737 000063 036604  PS3:    MOV      #'3,TSECX      ; TLM SECTOR 3...
2863 024402 012737 000207 003440  MOV      #<1_7.>!7,INIDA ;...IS CYL 1, SECT 7...
2864 024410 012737 000214 024560  MOV      #H<ERR!HCRC!OPI>,PSERR
2865 024416 012737 036676 024556  MOV      #EXPHCRC,PSETXT ;...AND HAS A BAD HEADER CRC.
2866 024424 000446                   BR      2$
2867
2868 024426 112737 000064 036604  PS4:    MOV      #'4,TSECX      ; TLM SECTOR 4...
2869 024434 012737 125252 003440  MOV      #<525_7.>!52,INIDA ;...IS CYL 525, SECT 52...
2870 024442 000432                   BR      1$
2871 ;...AND IS ERROR FREE.
2872 024444 112737 000065 036604  PS5:    MOV      #'5,TSECX      ; TLM SECTOR 5...

```

```

2873 024452 012737 135252 003440      MOV    #<565_7.>!52,INIDA ;...IS CYL 565, SECT 52...
2874 024460 000423                    BR     1$                ;...AND IS ERROR FREE.
2875
2876 024462                    PS7:   PRINTF #TURKY      ; THERE ISN'T ANY SECTOR 7...
(7) 024462 012746 024562                    MOV    #TURKY,-(SP)
(6) 024466 012746 000001                    MOV    #1,-(SP)
(3) 024472 010600                    MOV    SP,RO
(4) 024474 104417                    TRAP   C$PNTF
(4) 024476 062706 000004                    ADD    #4,SP
2877 024502 012737 000006 003314      MOV    #6,SNGLSEC      ;...CHANGE IT TO 6...
2878 024510 162700 000002      SUB    #2,RO          ;...ADJUST INDEX AND FALL THRU.
2879
2880 024514 112737 000066 036604 PS6:   MOVB   #'6,TSECX      ; TLM SECTOR 6...
2881 024522 012737 155555 003440      MOV    #<666_7.>!155,INIDA ;...IS CYL 666, SECT 55, HEAD 1...
2882 024530 005037 024560      1$:   CLR    PSERR
2883 024534 012737 036742 024556      MOV    #EXPNON,PSETXT
2884 024542 016037 024644 024642 2$:   MOV    DTABL(RO),DHEAD ; SET HCRC TO SEARCH FOR.
2885 024550 004737 025246      JSR    PC,FINDSEC     ; GO FIND AND READ IT TO BUF1.
2886 024554 000207      RTS    PC             ; RETURN, "V" = 1 IF SECTOR NOT FOUND.
2887
2888
2889 024556 036742      PSETXT: EXPNON      ; EXPECTED ERROR STATUS FOR EACH...
2890 024560 000000      PSERR:  0          ;...IS RETURNED IN THESE LOCATIONS.
2891
2892 024562 047045 040445 042523 TURKY: .ASCIZ /#N#ASECTOR 7 DOESN'T EXIST, USING 6 INSTEAD#N/
024570 052103 051117 033440
024576 042040 042517 047123
024604 052047 042440 044530
024612 052123 020054 051525
024620 047111 020107 020066
024626 047111 052123 040505
024634 022504 000116

```

2893

.EVEN

2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915 024640 025376
2916 024642 006051
2917 024644 006051
2918 024646 044051
2919 024650 132051
2920 024652 174001
2921 024654 036400
2922 024656 113554
2923
2924 024660 005037 003306
2925 024664 005000
2926 024666 077001
2927 024670 104005
2928 024672 113702 025753
2929 024676 004737 025550
2930 024702 025633
2931 024704 004737 025504
2932 024710 113700 004366
2933 024714 001407
2934 024716 120027 000060
2935 024722 103763
2936 024724 120027 000067
2937 024730 101360
2938 024732 110002
2939 024734 110201
2940 024736 013705 003320
2941 024742 042701 177770
2942 024746 001403
2943 024750 062705 000010
2944 024754 077103
2945
2946 024756 110237 025753
2947 024762 012737 025000 000004
2948 024770 012715 000001
2949 024774 000240
2950 024776 000410

```

.SBTTL          TLM PROM DUMP UTILITY.
;
; ROUTINE TO DUMP TLM ROM CONTENTS.
; CALLED FROM "INIT" SECTION IF PDSW=1.
; GET UNIT NUMBER, SECTOR NUMBER, AND OUTPUT FORMAT FROM OPERATOR.
; <+C> = ABORT AND RESTART DRS.
; LOAD SUBROUTINE USES "BUF2" AS A 256. WORD RING BUFFER FOR SEARCHING.
; IF FOUND, SECTOR CONTENTS (160. WORDS) ARE THEN COPIED TO "BUF1".
; THE LOAD SUBROUTINE IS ALSO USED IN TLM TESTS 5 AND 6
; TO ESTABLISH EXPECTED DATA PATTERNS.
;
; PSUEDO-DISK (ROM) SECTOR FORMAT:
;
;          WORDS          FUNCTION
;-----
;          1 - 17          SECTOR PULSE (12), T3(2), HEADER PREAMBLE (3).
;          18 - 20          DISK ADDRESS (1), ZERO (1), HEADER CRC (1).
;          21 - 24          HEADER POSTAMBLE (1), DATA PREAMBLE (3).
;          25 - 153         DATA (128.), DATA CRC (1).
;          154 - 160        DATA POSTAMBLE (1), AND GAP (6).
;
; DMODE:  OCTMOD          ; DUMP MODE (OCTAL OR HEX).
; DHEAD:  006051          ; DESIRED HEADER CRC WORD (DEFAULT SECT 1)...
; DTABL:  006051 ; 0C29  ; ...SELECTED FROM THIS TABLE.
;         044051 ; 4829
;         132051 ; B429
;         174001 ; F801
;         036400 ; 3D00
;         113554 ; 976C
;
; ROMDUMP: CLR PDSW          ; CLEAR DUMP REQUEST.
;          CLR RO
;          SOB RO..          ; DELAY...
;          EMT+5             ; ...AND CLEAR KBD.
;          MOVH HUNIT,R2     ; GET DEFAULT UNIT.
;          JSR PC,KBDOUT     ; SAY "HELLO".
;          GO
;          JSR PC,KBDIN      ; GET A UNIT NUMBER.
;          MOVH BUF1,RO
;          BEQ 2$            ; IF NONE, USE DEFAULT (LAST).
;          CMPB RO,#'0
;          BLO 1$            ; BR IF INVALID UNIT.
;          CMPB RO,#'7
;          BHI 1$            ;
;          MOVH RO,R2        ; DITTO
;          MOVH R2,R1        ; SAVE NEW UNIT FOR A MOMENT.
;          MOV TCS0,R5       ; BASE CSR => R5.
;          BIC #+C7,R1       ; STRIP UNIT NUM...
;          BEQ 3$            ; ...AND BR IF ZERO.
;          ADD #10,R5        ; ELSE, ADJUST UP TO SELECTED UNIT.
;          SOB R1,22$
;
;          MOVH R2,HUNIT     ; OK, SET UNIT IN TEXT.
;          MOV #4$,ERRVEC    ; SET A TRAP CATCHER.
;          MOV #1,(R5)       ; RESET TLM...
;          240
;          BR 5$            ; ...AND PROCEED IF IT DIDN'T TRAP.

```

```

2951 025000 022626          4$:  CMP      (SP)+,(SP)+
2952 025002 113737 025753 025610  MOVB    HUNIT,NOANS+6
2953 025010 004737 025550          JSR      PC,KBDOUT      ; SELECTED UNIT DOESN'T ANSWER.
2954 025014 025602          NOANS
2955 025016 000720          BR       ROMDUMP
2956
2957 025020 004737 025550          5$:  JSR      PC,KBDOUT      ; GET A SECTOR NUMBER.
2958 025024 025664          Q1
2959 025026 004737 025504          JSR      PC,KBDIN
2960 025032 113700 004366          MOVB    BUF1,R0
2961 025036 001416          BEQ     6$              ; IF NONE, REDO LAST SECTOR.
2962 025040 120027 000060          CMPB   RO,#'0
2963 025044 103765          BLO    5$
2964 025046 120027 000067          CMPB   RO,#'7
2965 025052 101362          BHI    5$
2966 025054 110037 025766          MOVB   RO,HSECT      ; SET IN HEADER STRING.
2967 025060 042700 177770          BIC    #'C7,R0
2968 025064 006300          ASL    RO              ; SHIFT UP TO WORD OFFSET...
2969 025066 016037 024642 024642  MOV     DHEAD(R0),DHEAD ; ...AND SET HEADER TO SEARCH FOR.
2970
2971 025074 004737 025550          6$:  JSR      PC,KBDOUT      ; GET AN OUTPUT MODE.
2972 025100 025711          Q2
2973 025102 004737 025504          JSR      PC,KBDIN
2974 025106 113700 004366          MOVB    BUF1,R0
2975 025112 001411          BEQ     7$              ; IF NONE, USE CURRENT FORMAT.
2976 025114 012737 025376 024640  MOV     #OCTMOD,DMODE ; ASSUME OCTAL...
2977 025122 120027 000131          CMPB   RO,#'Y
2978 025126 001003          BNE    7$
2979 025130 012737 025422 024640  MOV     #HEXMOD,DMODE ; ...CHANGE TO HEX IF REQUIRED.
2980 025136
2981 025136 016537 000006 003424 7$:  DOIT:  MOV     6(R5),TMPO      ; GET PROM ID.
2982 025144 004737 025550          JSR      PC,KBDOUT      ; SHOW UNIT AND SECTOR...
2983 025150 025742          HEADR
2984 025152 012701 003424          MOV     #TMPO,R1      ; ...AND PROM ID...
2985 025156 004777 177456          JSR      PC,@DMODE      ; ...IN THE CURRENT CONTEXT.
2986 025162 104007          EMT+7 ; <CRLF>
2987 025164 004737 025246          JSR      PC,FINDSEC     ; SEARCH FOR AND READ SECTOR.
2988 025170 102004          BVC    1$              ; BR IF SECTOR FOUND.
2989 025172 004737 025550          JSR      PC,KBDOUT
2990 025176 026003          CANTFIND
2991 025200 000627          BR       ROMDUMP
2992
2993 025202 012701 004366          1$:  MOV     #BUF1,R1      ; NOW FOUND SECTOR IS IN BUF1.
2994 025206 012702 000240          MOV     #160.,R2      ; DUMP 160. WORDS AT 8 PER LINE.
2995 025212 104007          EMT+7 ; <CRLF>
2996 025214 004737 025550          2$:  JSR      PC,KBDOUT
2997 025220 026030          TAB
2998 025222 004777 177412          JSR      PC,@DMODE      ; PRINT FROM (R1) IN OCTAL OR HEX.
2999 025226 005302          DEC    R2
3000 025230 032702 000007          BIT    #7,R2
3001 025234 001367          BNE    2$              ; LOOP 'TIL LINE DONE...
3002 025236 104007          EMT+7 ; <CRLF>
3003 025240 005702          TST    R2
3004 025242 001364          BNE    2$              ; ...AND LOOP 'TIL ALL DONE.
3005 025244 000605          BR       ROMDUMP
3006

```

```

3007 ; SEARCH FOR SELECTED SECTOR AND LOAD 160. WORD BLOCK TO BUF1.
3008 ; IF UNSUCCESSFUL, RETURN TO CALLER WITH "V" BIT SET.
3009 ;
3010 FINDSEC: MOV #1,(R5) ; RESET TLM...
3011 MOV #34,(R5) ; ...AND SET PROM READ MODE.
3012 CLR -(SP) ; CLEAR "FOUND" FLAG.
3013 CLR R0 ; SET LOOP CONTROL...
3014 MOV #BUFEND,R1 ; NOW, FILL BUF2 FROM TOP DOWN.
3015 1$: CMP R1,#BUF2 ; IN LOOP -- RESET RING POINTER...
3016 BNE 2$ ; ...WHEN IT REACHES THE BOTTOM.
3017 MOV #BUFEND,R1
3018 2$: INC 6(R5) ; INCR ROM ADDRESS...
3019 MOV 6(R5),-(R1) ; ...AND GET A WORD.
3020 CMP (R1),DHEAD ; IS THIS THE DESIRED HCRC WORD ??
3021 BNE 3$ ; BR IF NOT.
3022 INC (SP) ; YES, SET "FOUND" FLAG...
3023 MOV #20.,R0 ; ...AND SET TO READ 19 MORE.
3024 3$: SOB R0,1$ ; LOOP 'TIL DONE...
3025 MOV #1,(R5) ; ...THEN, RESET TLM...
3026 MOV #200,(R5) ; ...AND RESTORE DRIVE READY.
3027 TST (SP)+ ; SECTOR FOUND ??
3028 BNE 4$ ; BR IF SO.
3029 SEV ; NO, SET "V" BIT...
3030 RTS PC ; ...AND RETURN.
3031
3032 4$: MOV #BUF1,R2 ; NOW COPY FOUND SECTOR TO BUF1.
3033 MOV #160.,R3
3034 5$: CMP R1,#BUFEND ; IN LOOP -- RESET RING POINTER...
3035 BNE 6$ ; ...WHEN IT REACHES THE TOP.
3036 MOV #BUF2,R1
3037 6$: MOV (R1)+,(R2)+ ; BUF2 => BUF1.
3038 SOB R3,5$
3039 RTS PC ; RETURN, "V" IS CLEAR.
3040
3041 ; SUBROUTINES TO DUMP DATA IN OCTAL OR HEX.
3042 ;
3043 OCTMOD: PRINTF #OCTTXT,(R1)+
3044 (8) 025376 012146 MOV (R1)+,-(SP)
3045 (7) 025400 012746 026042 MOV #OCTTXT,-(SP)
3046 (6) 025404 012746 000002 MOV #2,-(SP)
3047 (3) 025410 010600 MOV SP,R0
3048 (4) 025412 104417 TRAP C$PNTF
3049 (4) 025414 062706 000006 ADD #6,SP
3044 025420 000207 RTS PC
3045
3046 HEXMOD: MOV R5,-(SP) ; SAVE TLM POINTER.
3047 1$: MOV #HEXTXT,R0 ; SET HEX TEXT POINTER.
3048 MOV (R1)+,R4 ; NEXT WORD => R4.
3049 2$: CLR R5 ; ...AND PARSE EACH NIBBLE IN R5.
3050 MOV #4,R3
3051 3$: ASL R4
3052 ROL R5 ; NIBBLE => R5<3:0>
3053 SOB R3,3$
3054 ADD #'0,R5 ; CONVERT TO ASCII 0 TO 9...
3055 CMPB R5,#'9
3056 BLE 4$

```



```

3057 025460 062705 000007          ADD      #7,R5          ;...AND A(10) TO F(15)...
3058 025464 110520          4$:   MOVB    R5,(R0)+    ;...AND STUFF IT.
3059 025466 105710          TSTB   (R0)
3060 025470 001360          BNE    2$
3061 025472 004737 025550      JSR    PC,KBDOUT      ; LOOP 'TIL 4 CHARS ASSEMBLED...
3062 025476 026035          HEXTXT
3063 025500 012605          MOV    (SP)+,R5      ; RESTORE R5.
3064 025502 000207          RTS    PC
3065
3066 025504 012701 004366      KBDIN: MOV    #BUF1,R1  ; TEMP ASCII BUFFER.
3067 025510 104005          1$:   EMT+5          ; CHAR => R0.
3068 025512 110011          MOVB   R0,(R1)
3069 025514 001775          BEQ    1$
3070 025516 120027 000003      CMPB   R0,#3
3071 025522 001411          BEQ    3$            ; EXIT ON <↑C>.
3072 025524 120027 000015      CMPB   R0,#15
3073 025530 001403          BEQ    2$            ; BR ON TERMINATOR.
3074 025532 104004          EMT+4          ; OTHERWISE, ECHO...
3075 025534 005201          INC    R1          ;...BUMP POINTER...
3076 025536 000764          BR     1$          ;...AND LOOP.
3077 025540 105011          2$:   CLRB   (R1)      ; TERMINATE INPUT STRING...
3078 025542 104007          EMT+7          ;...ECHO CRLF...
3079 025544 000207          RTS    PC          ;...AND RETURN TO CALLER.
3080 025546          3$:   DOCLN
(3) 025546 104444          TRAP   C$DCLN
3081
3082          KBDOUT: PRINTF #PURE,@(SP)
(8) 025550 017646 000000          MOV    @(SP),-(SP)
(7) 025554 012746 026046          MOV    #PURE, -(SP)
(6) 025560 012746 000002          MOV    #2, -(SP)
(3) 025564 010600          MOV    SP,R0
(4) 025566 104417          TRAP  C$PNTF
(4) 025570 062706 000006          ADD    #6,SP
3083 025574 062716 000002          ADD    #2,(SP)
3084 025600 000207          RTS    PC
3085
3086 025602 020040 046124 035115      NOANS: .ASCIZ / TLM:0 DOESN'T ANSWER/<15><12>
      025610 020060 047504 051505
      025616 023516 020124 047101
      025624 053523 051105 005015
      025632 000
3087 025633 015 012          Q0:   .ASCII <15><12>
3088 025635 040 052040 046514      .ASCIZ / TLM UNIT (0 TO 7) ? /
      025642 052440 044516 020124
      025650 030050 052040 020117
      025656 024467 037411 000040
3089 025664 020040 042523 052103      Q1:   .ASCIZ / SECTOR (1 TO 6) ? /
      025672 051117 024040 020061
      025700 047524 033040 004451
      025706 020077 000
3090 025711 040 044040 054105      Q2:   .ASCIZ / HEX OUTPUT (Y OR N) ? /
      025716 047440 052125 052520
      025724 020124 054450 047440
      025732 020122 024516 037411
      025740 000040
3091 025742 005015 052411 044516      HEADR: .ASCII <15><12><11>/UNIT: /

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 44-9
TLM PROM DUMP UTILITY.

SEQ 0110

3092	025750	035124	040			
	025753	060	020040	042523	HUNIT: .ASCII	/0 SECTOR: /
	025760	052103	051117	020072		
3093	025766	020061	050040	047522	HSECT: .ASCIZ	/1 PROM ID: /
	025774	020115	042111	020072		
	026002	000				
3094	026003	011	040503	023516	CANTFIND: .ASCIZ <11>/CAN'T FIND HEADER/<15><12>	
	026010	020124	044506	042116		
	026016	044040	040505	042504		
	026024	006522	000012			
3095	026030	000011			TAB: .ASCIZ	<11>
3096	026032	005015	000		CRLF: .ASCIZ	<15><12>
3097	026035	101	041502	000104	HEXTXT: .ASCIZ	/ABCD/
3098	026042	047445	000066		OCTTXT: .ASCIZ	/#06/
3099	026046	052045	000		PURE: .ASCIZ	/#T/
3100		026052				.EVEN

3102
3103
3104
3105
3106
3107 026052 013700 003402
3108 026056 001414
3109 026060 020027 000002
3110 026064 103406
3111 026066 005737 003344
3112 026072 001403
3113 026074 012700 000043
3114 026100 000402
3115 026102 012700 000035
3116 026106 000402
3117 026110 012700 000026
3118 026114 010037 026122
3119 026120 000207

```

.SBTTL
.SBTTL SUPERVISOR DISPATCH TABLE.
; SUBROUTINE TO ADJUST THE NUMBER OF TESTS TO RUN IAW CONTROLLER TYPE.
;
ADJTN: MOV RL TYP,RO ; GET RL TYPE.
      BEQ 2$ ; BR IF RL11.
      CMP RO,#RLV12
      BLO 1$ ; BR IF RLV11.
      TST TCSR
      BEQ 1$ ; BR IF TLM BYPASSED.
      MOV #35.,RO ; RLV12 WITH TLM RUNS ALL.
      SKP2
1$: MOV #29.,RO ; RLV12 OR RLV11 RUN THRU 29.
      SKP2
2$: MOV #22.,RO ; RL11 RUN THRU 22.
      MOV RO,L$DISPATCH-2 ; ...AND SET LAST TEST NUMBER.
      RTS PC

```

DISPATCH T\$TESTNUM

3120
3121 026122
(4) 026122 000043
(3) 026124
(6) 026124 012172
(6) 026126 012256
(6) 026130 012342
(6) 026132 012426
(6) 026134 012512
(6) 026136 012634
(6) 026140 013076
(6) 026142 013216
(6) 026144 013322
(6) 026146 013410
(6) 026150 013514
(6) 026152 013742
(6) 026154 014126
(6) 026156 014274
(6) 026160 014466
(6) 026162 014670
(6) 026164 015114
(6) 026166 015340
(6) 026170 015630
(6) 026172 016024
(6) 026174 016232
(6) 026176 016320
(6) 026200 016466
(6) 026202 016622
(6) 026204 016706
(6) 026206 017034
(6) 026210 017734
(6) 026212 020360
(6) 026214 020554
(6) 026216 021350
(6) 026220 021722
(6) 026222 022126
(6) 026224 022274
(6) 026226 022554

```

      .WORD 35
L$DISPATCH:
      .WORD T1
      .WORD T2
      .WORD T3
      .WORD T4
      .WORD T5
      .WORD T6
      .WORD T7
      .WORD T8
      .WORD T9
      .WORD T10
      .WORD T11
      .WORD T12
      .WORD T13
      .WORD T14
      .WORD T15
      .WORD T16
      .WORD T17
      .WORD T18
      .WORD T19
      .WORD T20
      .WORD T21
      .WORD T22
      .WORD T23
      .WORD T24
      .WORD T25
      .WORD T26
      .WORD T27
      .WORD T28
      .WORD T29
      .WORD T30
      .WORD T31
      .WORD T32
      .WORD T33
      .WORD T34

```


CVRLB80 -- RLV12 DISKLESS.
CVRLB8.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 45-1
SUPERVISOR DISPATCH TABLE.

I9

SEQ 0112

(6) 026230 023436

.WORD T35

J9

```

3123      .SBTTL GLOBAL ERROR HANDLERS AND ASCII TEXT
3124      ;
3125      ; THESE ARE THE HANDLERS FOR THE VARIOUS ERROR SIGNATURES.
3126      ;
3127      BGNMSG ERRSIG
(3) 026232 000137 027360      NOSIG: JMP CKERLT      ; SOME HAVE NO SIGNATURE.
3128      026232 000137 027360      ERRSIG::
3129      026236 000137 027152      ERRO:  JMP ALLREGS      ; ALL REGISTERS.
3130      026242 000137 027152      ERR1:  PRINTB #FRMT2,GDDAT ; BUS-TIMEOUT REG ADDR.
3131      (8) 026242 013746 003432      MOV      GDDAT,-(SP)
3132      (7) 026246 012746 027553      MOV      #FRMT2,-(SP)
3133      (6) 026252 012746 000002      MOV      #2,-(SP)
3134      (3) 026256 010600      MOV      SP,RO
3135      (4) 026260 104414      TRAP    C$PNTB
3136      (4) 026262 062706 000006      ADD     #6,SP
3137      026266 000137 027360      JMP     CKERLT
3138      026272 000137 027360      ERR2:  PRINTB #FRMT4,GDDAT,BDDAT ; EXP'D VS REC'D.
3139      (9) 026272 013746 003434      MOV      BDDAT,-(SP)
3140      (8) 026276 013746 003432      MOV      GDDAT,-(SP)
3141      (7) 026302 012746 027654      MOV      #FRMT4,-(SP)
3142      (6) 026306 012746 000003      MOV      #3,-(SP)
3143      (3) 026312 010600      MOV      SP,RO
3144      (4) 026314 104414      TRAP    C$PNTB
3145      (4) 026316 062706 000010      ADD     #10,SP
3146      026322 000137 027360      JMP     CKERLT
3147      026326 000137 027360      ERR3:  PRINTB #FRMT5,BDDAT      ; MAINT SEQ FAILED STATE.
3148      (8) 026326 013746 003434      MOV      BDDAT,-(SP)
3149      (7) 026332 012746 027710      MOV      #FRMT5,-(SP)
3150      (6) 026336 012746 000002      MOV      #2,-(SP)
3151      (3) 026342 010600      MOV      SP,RO
3152      (4) 026344 104414      TRAP    C$PNTB
3153      (4) 026346 062706 000006      ADD     #6,SP
3154      026352 000137 027152      JMP     ALLREGS
3155      026356 000137 027152      ERR4:  PRINTB #FRMT4,GDDAT,BDDAT ; EXP'D VS REC'D...
3156      (9) 026356 013746 003434      MOV      BDDAT,-(SP)
3157      (8) 026362 013746 003432      MOV      GDDAT,-(SP)
3158      (7) 026366 012746 027654      MOV      #FRMT4,-(SP)
3159      (6) 026372 012746 000003      MOV      #3,-(SP)
3160      (3) 026376 010600      MOV      SP,RO
3161      (4) 026400 104414      TRAP    C$PNTB
3162      (4) 026402 062706 000010      ADD     #10,SP
3163      026406 000137 027152      JMP     ALLREGS      ; ...AND REGISTERS.
3164      026412 000137 027152      ERR6:  PRINTB #FRMT3,#EM99      ; CS ERRORS.
3165      (8) 026412 012746 034456      MOV      #EM99,-(SP)
3166      (7) 026416 012746 027571      MOV      #FRMT3,-(SP)
3167      (6) 026422 012746 000002      MOV      #2,-(SP)
3168      (3) 026426 010600      MOV      SP,RO
3169      (4) 026430 104414      TRAP    C$PNTB
3170      (4) 026432 062706 000006      ADD     #6,SP
3171      026436 000137 027152      JMP     ALLREGS

```

```

3146
3147 026442          ERR7:  PRINTB  #FRMT6,TMPO      ; CPU PRIORITY.
(8) 026442 013746 003424
(7) 026446 012746 027742
(6) 026452 012746 000002
(3) 026456 010600
(4) 026460 104414
(4) 026462 062706 000006
3148 026466 000137 027360          JMP      CKERLT
3149
3150 026472          ERR10: PRINTB  #FRMT3A,#EM99   ; EXPECTED OPI ERRORS.
(8) 026472 012746 034456
(7) 026476 012746 027613
(6) 026502 012746 000002
(3) 026506 010600
(4) 026510 104414
(4) 026512 062706 000006
3151 026516 000137 027152          JMP      ALLREGS
3152
3153 026522          ERR11: PRINTB  #FRMT10,BDDAT   ; OPI TIMING ERROR.
(8) 026522 013746 003434
(7) 026526 012746 027775
(6) 026532 012746 000002
(3) 026536 010600
(4) 026540 104414
(4) 026542 062706 000006
3154 026546 000137 027360          JMP      CKERLT
3155
3156 026552 004737 026662          ERR12: JSR      PC,PRTPA      ; STAT WRONG ON NXM (ABOVE 32K)...
3157 026556          PRINTB  #FRMT21,#EM99   ; ...OR BBS7/NXM FAILURE.
(8) 026556 012746 034456
(7) 026562 012746 030313
(6) 026566 012746 000002
(3) 026572 010600
(4) 026574 104414
(4) 026576 062706 000006
3158 026602 000137 027152          JMP      ALLREGS
3159
3160 026606 004737 026662          ERR13: JSR      PC,PRTPA      ; UNEX STAT ERRORS (ABOVE 32K).
3161 026612 000137 026412          JMP      ERR6
3162
3163 026616 004737 026662          ERR14: JSR      PC,PRTPA      ; DATA WRONG (ABOVE 32K).
3164 026622          PRINTB  #FRMT22,TMPO,GDDAT,BDDAT ; EXPD VS RECD.
(10) 026622 013746 003434
(9) 026626 013746 003432
(8) 026632 013746 003424
(7) 026636 012746 030350
(6) 026642 012746 000004
(3) 026646 010600
(4) 026650 104414
(4) 026652 062706 000012
3165 026656 000137 027152          JMP      ALLREGS
3166
3167 ; PACK AND PRINT 22 BIT PHYSICAL ADDRESS FROM TMP1.,TMP2.
3168 ;
3169 026662 005737 003430          PRTPA: TST      TMP2          ; BIT 15 ON ??

```



```

3170 026666 100003          BPL      1$          ; NO, "C" = 0.
3171 026670 062737 100000 003430  ADD      %BIT15,TMP2 ; YES, "C" = 1.
3172 026676 006137 003426 1$:      ROL      TMP1          ; TMP1 = ADDR<21:15>...
3173 026702          PRINTB  #FRMT20,TMP1,TMP2 ; TMP2 = ADDR<14:00>.
(9) 026702 013746 003430          MOV      TMP2,-(SP)
(8) 026706 013746 003426          MOV      TMP1,-(SP)
(7) 026712 012746 030253          MOV      #FRMT20,-(SP)
(6) 026716 012746 000003          MOV      #3,-(SP)
(3) 026722 010600          MOV      SP,RO
(4) 026724 104414          TRAP    C$PNTB
(4) 026726 062706 000010          ADD      #10,SP
3174 026732 000207          RTS      PC
3175          ;
3176          ; TLM UNIQUE ERROR SIGNATURES.
3177          ;
3178 026734 004737 027114  ERR20: JSR      PC,E2022 ; BAD STATUS AFTER READ,READNH,WRTCHK...
3179 026740          ERR20A: ; ...OR AFTER SEEK,GET STATUS,WRITE.
3180 026740          PRINTX  #FRMT26,TMP1,#EM99
(9) 026740 012746 034456          MOV      #EM99,-(SP)
(8) 026744 013746 003426          MOV      TMP1,-(SP)
(7) 026750 012746 030462          MOV      #FRMT26,-(SP)
(6) 026754 012746 000003          MOV      #3,-(SP)
(3) 026760 010600          MOV      SP,RO
(4) 026762 104415          TRAP    C$PNTX
(4) 026764 062706 000010          ADD      #10,SP
3181 026770          PRINTX  #FRMT27          ; CRLF.
(7) 026770 012746 030503          MOV      #FRMT27,-(SP)
(6) 026774 012746 000001          MOV      #1,-(SP)
(3) 027000 010600          MOV      SP,RO
(4) 027002 104415          TRAP    C$PNTX
(4) 027004 062706 000004          ADD      #4,SP
3182 027010 000137 027152          JMP      ALLREGS
3183          ;
3184 027014 004737 027114  ERR21: JSR      PC,E2022 ; READ HEADER FAILS.
3185 027020 000137 027360          JMP      CKERLT
3186          ;
3187 027024 004737 027114  ERR22: JSR      PC,E2022 ; DATA FROM TLM WRONG.
3188 027030          PRINTX  #FRMT14,TMP1,TMPO,GDDAT,BDDAT
(11) 027030 013746 003434          MOV      BDDAT,-(SP)
(10) 027034 013746 003432          MOV      GDDAT,-(SP)
(9) 027040 013746 003424          MOV      TMPO,-(SP)
(8) 027044 013746 003426          MOV      TMP1,-(SP)
(7) 027050 012746 030103          MOV      #FRMT14,-(SP)
(6) 027054 012746 000005          MOV      #5,-(SP)
(3) 027060 010600          MOV      SP,RO
(4) 027062 104415          TRAP    C$PNTX
(4) 027064 062706 000014          ADD      #14,SP
3189 027070          PRINTX  #FRMT27          ; CRLF
(7) 027070 012746 030503          MOV      #FRMT27,-(SP)
(6) 027074 012746 000001          MOV      #1,-(SP)
(3) 027100 010600          MOV      SP,RO
(4) 027102 104415          TRAP    C$PNTX
(4) 027104 062706 000004          ADD      #4,SP
3190 027110 000137 027152          JMP      ALLREGS
3191          ;
3192 027114          E2022: PRINTX  #FRMT25,#TSEC,TMP2,#TPEAK

```

```

(10) 027114 012746 036654
(9) 027120 013746 003430
(8) 027124 012746 036575
(7) 027130 012746 030453
(6) 027134 012746 000004
(3) 027140 010600
(4) 027142 104415
(4) 027144 062706 000012
3193 027150 000207
3194
3195
3196
3197
3198 027152 005000
3199 027154 023727 003402 000003
3200 027162 001002
3201 027164 012700 000045
3202 027170 110037 027477
3203 027174 110037 027541
3204 027200
(7) 027200 012746 027440
(6) 027204 012746 000001
(3) 027210 010600
(4) 027212 104414
(4) 027214 062706 000004
3205 027220
(13) 027220 013746 003360
(12) 027224 013746 003356
(11) 027230 013746 003354
(10) 027234 013746 003352
(9) 027240 013746 003350
(8) 027244 012746 030506
(7) 027250 012746 027510
(6) 027254 012746 000007
(3) 027260 010600
(4) 027262 104414
(4) 027264 062706 000020
3206 027270
(13) 027270 013746 003376
(12) 027274 013746 003372
(11) 027300 013746 003370
(10) 027304 013746 003366
(9) 027310 013746 003364
(8) 027314 012746 030527
(7) 027320 012746 027510
(6) 027324 012746 000007
(3) 027330 010600
(4) 027332 104414
(4) 027334 062706 000020
3207 027340
(7) 027340 012746 027550
(6) 027344 012746 000001
(3) 027350 010600
(4) 027352 104414
(4) 027354 062706 000004
3208

```

```

RTS PC
;
; DISPLAY ALL REGISTERS BEFORE AND AFTER ERROR (IF REQ'D),
; CHECK IF ERROR LIMIT EXCEEDED AND EXIT ACCORDINGLY.
;
ALLREGS: CLR R0 ; DISABLE...
          CMP RLTP,#RLV12X
          BNE 1$
          MOV #'%,R0 ; ...OR ENABLE THE EXTRA REGISTER.
          MOVB RO,FRMT1A ; ENABLE/DISABLE BAE TEXT.
          MOVB RO,FRMT1C
          PRINTB #FRMT1 ; HEADER.
;
          PRINTB #FRMT1B,#BEREG,B.CS,B.BA,B.DA,B.MP,B.BAE
;
          PRINTB #FRMT1B,#AFREG,E.CS,E.BA,E.DA,E.MP,E.BAE
;
          PRINTB #FRMT1D
; FALL THRU...

```

```

MOV #TPEAK,-(SP)
MOV TMP2,-(SP)
MOV #TSEC,-(SP)
MOV #FRMT25,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTX
ADD #12,SP

MOV #FRMT1,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP

MOV B.BAE,-(SP)
MOV B.MP,-(SP)
MOV B.DA,-(SP)
MOV B.BA,-(SP)
MOV B.CS,-(SP)
MOV #BEREG,-(SP)
MOV #FRMT1B,-(SP)
MOV #7,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #20,SP

MOV E.BAE,-(SP)
MOV E.MP,-(SP)
MOV E.DA,-(SP)
MOV E.BA,-(SP)
MOV E.CS,-(SP)
MOV #AFREG,-(SP)
MOV #FRMT1B,-(SP)
MOV #7,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #20,SP

MOV #FRMT1D,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP

```

```

3209      ;
3210      ;...AND CHECK ERROR LIMIT (IF ANY).
3211      ;
3212      CKERLT: INLOOP
(3) 027360 104420                                TRAP    C$INLP
3213      BCOMPLETE 3$                            ; RETURN IF LOOPING...
(2) 027362 103425                                BCS     3$
3214      027364 005737 003304                    TST     ERRLMT
3215      027370 001422                            BEQ     3$
3216      027372 005277 154070                    INC     @ERPOINT
3217      027376 027737 154064 003304            CMP     @ERPOINT,ERRLMT
3218      027404 002414                            BLT     3$
3219      027406                                PRINTF  @FRMT23
(7) 027406 012746 030422                            MOV     @FRMT23,-(SP)
(6) 027412 012746 000001                            MOV     #1,-(SP)
(3) 027416 010600                                MOV     SP,RO
(4) 027420 104417                                TRAP   C$PNTF
(4) 027422 062706 000004                            ADD     #4,SP
3220      027426                                2$:    DODU  UNITST
(3) 027426 013700 003322                            ;DROP THE UNIT...
(3) 027432 104451                                MOV     UNITST,RO
3221      DOCLN                                    TRAP   C$DODU
(3) 027434 104444                                ;...AND ABORT.
3222      027436                                3$:    ENDMSG
(3) 027436                                ; OR RETURN TO CALLER.
(3) 027436 104423                                L10063: TRAP   C$MSG

```



```

3224
3225      ; FORMATTED ASCII TEXT.
3226      ;
3227 027440 040445 042522 044507 FRMT1: .ASCII /%AREGISTERS CS BA DA MP/
      027446 052123 051105 004523
      027454 020040 051503 020011
      027462 041040 004501 020040
      027470 040504 020011 046440
      027476 120
3228 027477 045 004501 020040 FRMT1A: .ASCIZ /%A BAE/
      027504 040502 000105
3229 027510 047045 052045 047445 FRMT1B: .ASCII /%N%T%06%S2%06%S2%06%S2%06/
      027516 022466 031123 047445
      027524 022466 031123 047445
      027532 022466 031123 047445
      027540 066
3230 027541 045 031123 047445 FRMT1C: .ASCIZ /%S2%06/
      027546 000066
3231 027550 047045 000 FRMT1D: .ASCIZ /%N/
3232 027553 045 040501 042104 FRMT2: .ASCIZ /%AADDR: %06%N/
      027560 035122 022440 033117
      027566 047045 000
3233 027571 045 042501 051122 FRMT3: .ASCIZ /%AERRORS SET:%T%N/
      027576 051117 020123 042523
      027604 035124 052045 047045
      027612 000
3234 027613 045 042501 050130 FRMT3A: .ASCIZ /%AEXP'D: ERR HNF OPI REC'D:%T%N/
      027620 042047 020072 051105
      027626 020122 047110 020106
      027634 050117 020111 051040
      027642 041505 042047 022472
      027650 022524 000116
3235 027654 040445 054105 023520 FRMT4: .ASCIZ /%AEXP'D: %06%A REC'D: %06%N/
      027662 035104 022440 033117
      027670 040445 051040 041505
      027676 042047 020072 047445
      027704 022466 000116
3236 027710 040445 052101 044440 FRMT5: .ASCIZ /%AAT INTERNAL STATE %D2%N/
      027716 052116 051105 040516
      027724 020114 052123 052101
      027732 020105 042045 022462
      027740 000116
3237 027742 040445 052101 050040 FRMT6: .ASCIZ /%AAT PROCESSOR LEVEL %01%N/
      027750 047522 042503 051523
      027756 051117 046040 053105
      027764 046105 022440 030517
      027772 047045 000
3238 027775 045 047501 044520 FRMT10: .ASCII /%AOPI FLAG RECEIVED AT%D4%A MSEC.%N/
      030002 043040 040514 020107
      030010 042522 042503 053111
      030016 042105 040440 022524
      030024 032104 040445 046440
      030032 042523 027103 047045
3239 030040 040445 054105 042520 .ASCIZ /%AEXPECTED FROM 155 TO 650 MSEC.%N/
      030046 052103 042105 043040
      030054 047522 020115 032461

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 46-6
GLOBAL ERROR HANDLERS AND ASCII TEXT

SEQ 0119

	030062	020065	047524	033040	
	030070	030065	046440	042523	
	030076	027103	047045	000	
3240	030103	045	022516	053501	FRMT14: .ASCIZ /%N%AWORD: %03%A ADDR: %06%A EXP'D: %06%A REC'D: %06/
	030110	051117	035104	022440	
	030116	031517	040445	020040	
	030124	042101	051104	020072	
	030132	047445	022466	020101	
	030140	042440	050130	042047	
	030146	020072	047445	022466	
	030154	020101	051040	041505	
	030162	042047	020072	047445	
	030170	000066			
3241	030172	047045	042045	022463	FRMT15: .ASCIZ /%N%D3%A WORDS BAD OUT OF 255 WORDS TRANSFERRED%N/
	030200	020101	047527	042122	
	030206	020123	040502	020104	
	030214	052517	020124	043117	
	030222	031040	032465	053440	
	030230	051117	051504	052040	
	030236	040522	051516	042506	
	030244	051122	042105	047045	
	030252	000			
3242	030253	045	040501	020124	FRMT20: .ASCIZ /%AAT PHYSICAL ADDRESS: %03%05%N/
	030260	044120	051531	041511	
	030266	046101	040440	042104	
	030274	042522	051523	020072	
	030302	047445	022463	032517	
	030310	047045	000		
3243	030313	045	042501	050130	FRMT21: .ASCIZ /%AEXP'D: ERR NXM REC'D:%T%N/
	030320	042047	020072	051105	
	030326	020122	054116	020115	
	030334	051040	041505	042047	
	030342	022472	022524	000116	
3244	030350	040445	047527	042122	FRMT22: .ASCIZ /%AWORD: %03%A EXP'D: %06%A REC'D: %06%N/
	030356	020072	047445	022463	
	030364	020101	042440	050130	
	030372	042047	020072	047445	
	030400	022466	020101	051040	
	030406	041505	042047	020072	
	030414	047445	022466	000116	
3245	030422	040445	051105	047522	FRMT23: .ASCIZ /%AERROR LIMIT EXCEEDED%N/
	030430	020122	044514	044515	
	030436	020124	054105	042503	
	030444	042105	042105	047045	
	030452	000			
3246					
3247	030453	045	022524	022524	FRMT25: .ASCIZ /%T%T%T/
	030460	000124			
3248	030462	047045	052045	040445	FRMT26: .ASCIZ /%N%T%A REC'D:%T/
	030470	020040	042522	023503	
	030476	035104	052045	000	
3249	030503	045	000116		FRMT27: .ASCIZ /%N/
3250					.EVEN
3251					;
3252					; PURE (UNFORMATTED) ASCII TEXT.
3253					;

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 46-7
GLOBAL ERROR HANDLERS AND ASCII TEXT

SEQ 0120

3254	030506	042502	047506	042522	BEREG: .ASCIZ /BEFORE COMMAND: /
	030514	041440	046517	040515	
	030522	042116	020072	000	
3255	030527	101	052106	051105	AFREG: .ASCIZ /AFTER COMMAND: /
	030534	041440	046517	040515	
	030542	042116	020072	000040	
3256	030550	047503	052116	047522	CRTIM: .ASCIZ /CONTROLLER TIMED OUT/
	030556	046114	051105	052040	
	030564	046511	042105	047440	
	030572	052125	000		
3257	030575	104	044522	042526	DRTIM: .ASCIZ /DRIVE READY TIMED OUT/
	030602	051040	040505	054504	
	030610	052040	046511	042105	
	030616	047440	052125	000	
3258	030623	040	051105	000122	CERR: .ASCIZ / ERR/
3259	030630	042040	053122	000	DEMES: .ASCIZ / DRV/
3260	030635	040	054116	000115	NXMMES: .ASCIZ / NXM/
3261	030642	050040	051101	000	PARMES: .ASCIZ / PAR/
3262	030647	040	046104	000124	DLTMES: .ASCIZ / DLT/
3263	030654	044040	043116	000	HNFMES: .ASCIZ / HNF/
3264	030661	040	041504	041522	DCRCMES: .ASCIZ / DCRC/
	030666	000			
3265	030667	040	041510	041522	HCRCMES: .ASCIZ / HCRC/
	030674	000			
3266	030675	040	050117	000111	OPIMES: .ASCIZ / OPI/
3267	030702	047040	047117	000105	NON: .ASCIZ / NONE/
3268					
3269	030710	047503	052116	047522	EMO: .ASCIZ /CONTROLLER DOESN'T ANSWER -- CAN'T TEST/
	030716	046114	051105	042040	
	030724	042517	047123	052047	
	030732	040440	051516	042527	
	030740	020122	026455	041440	
	030746	047101	052047	052040	
	030754	051505	000124		
3270	030760	040503	020116	047516	EM1: .ASCIZ /CAN NOT ADDRESS RLCS/
	030766	020124	042101	051104	
	030774	051505	020123	046122	
	031002	051503	000		
3271	031005	103	047101	047040	EM2: .ASCIZ /CAN NOT ADDRESS RLBA/
	031012	052117	040440	042104	
	031020	042522	051523	051040	
	031026	041114	000101		
3272	031032	040503	020116	047516	EM3: .ASCIZ /CAN NOT ADDRESS RLDA/
	031040	020124	042101	051104	
	031046	051505	020123	046122	
	031054	040504	000		
3273	031057	103	047101	047040	EM4: .ASCIZ /CAN NOT ADDRESS RLMP/
	031064	052117	040440	042104	
	031072	042522	051523	051040	
	031100	046514	000120		
3274	031104	040503	020116	047516	EM4A: .ASCIZ /CAN NOT ADDRESS RLBAE/
	031112	020124	042101	051104	
	031120	051505	020123	046122	
	031126	040502	000105		
3275	031132	040502	020105	044123	EM4B: .ASCIZ /BAE SHOULD BE DISABLED -- IT'S NOT !!!/
	031140	052517	042114	041040	

	031146	020105	044504	040523		
	031154	046102	042105	026440		
	031162	020055	052111	051447		
	031170	047040	052117	020440		
	031176	020441	000			
3276	031201	122	041514	020123	EM5:	.ASCIZ \RLCS READ/WRITE ERROR (BIT 0 DON'T CARE)\
	031206	042522	042101	053457		
	031214	044522	042524	042440		
	031222	051122	051117	024040		
	031230	044502	020124	020060		
	031236	047504	023516	020124		
	031244	040503	042522	000051		
3277	031252	046122	040502	051040	EM6:	.ASCIZ \RLBA READ/WRITE ERROR\
	031260	040505	027504	051127		
	031266	052111	020105	051105		
	031274	047522	000122			
3278	031300	046122	040504	051040	EM7:	.ASCIZ \RLDA READ/WRITE ERROR\
	031306	040505	027504	051127		
	031314	052111	020105	051105		
	031322	047522	000122			
3279	031326	046122	040502	020105	EM8:	.ASCIZ \RLBAE READ/WRITE ERROR\
	031334	042522	042101	053457		
	031342	044522	042524	042440		
	031350	051122	051117	000		
3280	031355	122	041114	020101	EM10:	.ASCIZ /RLBA ERROR AFTER MAINT. FUNCTION/
	031362	051105	047522	020122		
	031370	043101	042524	020122		
	031376	040515	047111	027124		
	031404	043040	047125	052103		
	031412	047511	000116			
3281	031416	046122	040504	042440	EM12:	.ASCIZ /RLDA ERROR AFTER MAINT. FUNCTION/
	031424	051122	051117	040440		
	031432	052106	051105	046440		
	031440	044501	052116	020056		
	031446	052506	041516	044524		
	031454	047117	000			
3282	031457	122	041514	020123	EM14:	.ASCIZ /RLCS INCORRECT AFTER NO-OP/
	031464	047111	047503	051122		
	031472	041505	020124	043101		
	031500	042524	020122	047516		
	031506	047455	000120			
3283	031512	047516	047455	020120	EM14A:	.ASCIZ /NO-OP ALTERED RLBA, RLDA, OR RLMP/
	031520	046101	042524	042522		
	031526	020104	046122	040502		
	031534	020054	046122	040504		
	031542	020054	051117	051040		
	031550	046514	000120			
3284	031554	040515	047111	042524	EM14B:	.ASCIZ /MAINTENANCE MODE SEQUENCER FAILED/
	031562	040516	041516	020105		
	031570	047515	042504	051440		
	031576	050505	042525	041516		
	031604	051105	043040	044501		
	031612	042514	000104			
3285	031616	047516	044440	052116	EM15:	.ASCIZ /NO INTERRUPT ON FUNCTION COMPLETE/
	031624	051105	052522	052120		
	031632	047440	020116	052506		

	031640	041516	044524	047117		
	031646	041440	046517	046120		
	031654	052105	000105			
3286	031660	047111	042524	051122	EM16:	.ASCIZ /INTERRUPT PRIORITY FAILURE/
	031666	050125	020124	051120		
	031674	047511	044522	054524		
	031702	043040	044501	052514		
	031710	042522	000			
3287	031713	106	051117	042503	EM17:	.ASCIZ /FORCED OPI -- INTERRUPT NOT RECEIVED/
	031720	020104	050117	020111		
	031726	026455	044440	052116		
	031734	051105	052522	052120		
	031742	047040	052117	051040		
	031750	041505	044505	042526		
	031756	000104				
3288	031760	051103	020103	043117	EM20:	.ASCIZ /CRC OF DA+3 INCORRECT (SERIAL DATA PATH)/
	031766	042040	025501	020063		
	031774	047111	047503	051122		
	032002	041505	020124	051450		
	032010	051105	040511	020114		
	032016	040504	040524	050040		
	032024	052101	024510	000		
3289	032031	103	041522	047440	EM21:	.ASCIZ /CRC OF CRC OF DA+4 INCORRECT (SERIAL DATA PATH)/
	032036	020106	051103	020103		
	032044	043117	042040	025501		
	032052	020064	047111	047503		
	032060	051122	041505	020124		
	032066	051450	051105	040511		
	032074	020114	040504	040524		
	032102	050040	052101	024510		
	032110	000				
3290	032111	106	046111	027514	EM22:	.ASCIZ \FILL/EMPTY FIFO DATA TRANSFER ERROR\
	032116	046505	052120	020131		
	032124	044506	047506	042040		
	032132	052101	020101	051124		
	032140	047101	043123	051105		
	032146	042440	051122	051117		
	032154	000				
3291	032155	106	046111	027514	EM23:	.ASCIZ \FILL/EMPTY FIFO LAST WORD+1 INCORRECT\
	032162	046505	052120	020131		
	032170	044506	047506	046040		
	032176	051501	020124	047527		
	032204	042122	030453	044440		
	032212	041516	051117	042522		
	032220	052103	000			
3292	032223	105	051122	051117	EM27:	.ASCII /ERROR STATUS INCORRECT AFTER FORCED OPI/<15><12>
	032230	051440	040524	052524		
	032236	020123	047111	047503		
	032244	051122	041505	020124		
	032252	043101	042524	020122		
	032260	047506	041522	042105		
	032266	047440	044520	005015		
3293	032274	047111	052111	040511		.ASCII /INITIAL WORD COUNT WAS /
	032302	020114	047527	042122		
	032310	041440	052517	052116		
	032316	053440	051501	040		

3294	032323	074	032440	030461	EM27X:	.ASCIZ	/< 511./
	032330	000056					
3295	032332	050117	020111	046106	EM31:	.ASCIZ	/OPI FLAG NOT RECEIVED/
	032340	043501	047040	052117			
	032346	051040	041505	044505			
	032354	042526	000104				
3296	032360	050117	020111	044524	EM32:	.ASCIZ	/OPI TIMING INCORRECT/
	032366	044515	043516	044440			
	032374	041516	051117	042522			
	032402	052103	000				
3297	032405	127	044522	044524	EM44:	.ASCIZ	/WRITING RLMP MODIFIED RLCS/
	032412	043516	051040	046514			
	032420	020120	047515	044504			
	032426	044506	042105	051040			
	032434	041514	000123				
3298	032440	051127	052111	047111	EM45:	.ASCIZ	/WRITING RLMP MODIFIED RLBA/
	032446	020107	046122	050115			
	032454	046440	042117	043111			
	032462	042511	020104	046122			
	032470	040502	000				
3299	032473	127	044522	044524	EM46:	.ASCIZ	/WRITING RLMP MODIFIED RLDA/
	032500	043516	051040	046514			
	032506	020120	047515	044504			
	032514	044506	042105	051040			
	032522	042114	000101				
3300	032526	051127	052111	047111	EM47:	.ASCIZ	/WRITING RLMP MODIFIED RLBAE/
	032534	020107	046122	050115			
	032542	046440	042117	043111			
	032550	042511	020104	046122			
	032556	040502	000105				
3301	032562	046122	051503	053440	EM50:	.ASCIZ	/RLCS WRONG AFTER WRITING RLBAE/
	032570	047522	043516	040440			
	032576	052106	051105	053440			
	032604	044522	044524	043516			
	032612	051040	041114	042501			
	032620	000					
3302	032621	122	041114	042501	EM51:	.ASCIZ	/RLBAE MODIFIED RLBA/
	032626	046440	042117	043111			
	032634	042511	020104	046122			
	032642	040502	000				
3303	032645	122	041114	042501	EM52:	.ASCIZ	/RLBAE MODIFIED RLDA/
	032652	046440	042117	043111			
	032660	042511	020104	046122			
	032666	040504	000				
3304	032671	042	044502	020124	EM61:	.ASCIZ	/"BIT SET" ON RLCS YIELDS WRONG RESULT/
	032676	042523	021124	047440			
	032704	020116	046122	051503			
	032712	054440	042511	042114			
	032720	020123	051127	047117			
	032726	020107	042522	052523			
	032734	052114	000				
3305	032737	042	044502	020124	EM62:	.ASCIZ	/"BIT CLEAR" ON RLCS YIELDS WRONG RESULT/
	032744	046103	040505	021122			
	032752	047440	020116	046122			
	032760	051503	054440	042511			
	032766	042114	020123	051127			

	032774	047117	020107	042522	
	033002	052523	052114	000	
3306	033007	042	044502	020124	EM63: .ASCIZ /"BIT SET" ON RLBA YIELDS WRONG RESULT/
	033014	042523	021124	047440	
	033022	020116	046122	040502	
	033030	054440	042511	042114	
	033036	020123	051127	047117	
	033044	020107	042522	052523	
	033052	052114	000		
3307	033055	042	044502	020124	EM64: .ASCIZ /"BIT CLEAR" ON RLBA YIELDS WRONG RESULT/
	033062	046103	040505	021122	
	033070	047440	020116	046122	
	033076	040502	054440	042511	
	033104	042114	020123	051127	
	033112	047117	020107	042522	
	033120	052523	052114	000	
3308	033125	042	044502	020124	EM65: .ASCIZ /"BIT SET" ON RLDA YIELDS WRONG RESULT/
	033132	042523	021124	047440	
	033140	020116	046122	040504	
	033146	054440	042511	042114	
	033154	020123	051127	047117	
	033162	020107	042522	052523	
	033170	052114	000		
3309	033173	042	044502	020124	EM66: .ASCIZ /"BIT CLEAR" ON RLDA YIELDS WRONG RESULT/
	033200	046103	040505	021122	
	033206	047440	020116	046122	
	033214	040504	054440	042511	
	033222	042114	020123	051127	
	033230	047117	020107	042522	
	033236	052523	052114	000	
3310	033243	042	044502	020124	EM67: .ASCIZ /"BIT SET" ON RLBAE YIELDS WRONG RESULT/
	033250	042523	021124	047440	
	033256	020116	046122	040502	
	033264	020105	044531	046105	
	033272	051504	053440	047522	
	033300	043516	051040	051505	
	033306	046125	000124		
3311	033312	041042	052111	041440	EM68: .ASCIZ /"BIT CLEAR" ON RLBAE YIELDS WRONG RESULT/
	033320	042514	051101	020042	
	033326	047117	051040	041114	
	033334	042501	054440	042511	
	033342	042114	020123	051127	
	033350	047117	020107	042522	
	033356	052523	052114	000	
3312	033363	102	051525	051055	EM69: .ASCIZ /BUS-RESET DIDN'T CLEAR RLCS/
	033370	051505	052105	042040	
	033376	042111	023516	020124	
	033404	046103	040505	020122	
	033412	046122	051503	000	
3313	033417	102	051525	051055	EM70: .ASCIZ /BUS-RESET DIDN'T CLEAR RLBA/
	033424	051505	052105	042040	
	033432	042111	023516	020124	
	033440	046103	040505	020122	
	033446	046122	040502	000	
3314	033453	102	051525	051055	EM71: .ASCIZ /BUS-RESET DIDN'T CLEAR RLDA/
	033460	051505	052105	042040	

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33MACY11 30(1046) 12-JUN-85 15:42 PAGE 46-12
GLOBAL ERROR HANDLERS AND ASCII TEXT

SEQ 0125

	033466	042111	023516	020124		
	033474	046103	040505	020122		
	033502	046122	040504	000		
3315	033507	102	051525	051055	EM71A:	.ASCIZ /BUS-RESET DIDN'T CLEAR RLBAE/
	033514	051505	052105	042040		
	033522	042111	023516	020124		
	033530	046103	040505	020122		
	033536	046122	040502	000105		
3316	033544	051127	052111	047111	EM72:	.ASCIZ /WRITING RLCS MODIFIED RLBA/
	033552	020107	046122	051503		
	033560	046440	042117	043111		
	033566	042511	020104	046122		
	033574	040502	000			
3317	033577	127	044522	044524	EM73:	.ASCIZ /WRITING RLCS MODIFIED RLDA/
	033604	043516	051040	041514		
	033612	020123	047515	044504		
	033620	044506	042105	051040		
	033626	042114	000101			
3318	033632	046122	040502	020105	EM73A:	.ASCIZ /RLBAE WRONG AFTER WRITING RLCS/
	033640	051127	047117	020107		
	033646	043101	042524	020122		
	033654	051127	052111	047111		
	033662	020107	046122	051503		
	033670	000				
3319	033671	127	044522	044524	EM74:	.ASCIZ /WRITING RLBA MODIFED RLCS/
	033676	043516	051040	041114		
	033704	020101	047515	044504		
	033712	042506	020104	046122		
	033720	051503	000			
3320	033723	127	044522	044524	EM75:	.ASCIZ /WRITING RLBA MODIFED RLDA/
	033730	043516	051040	041114		
	033736	020101	047515	044504		
	033744	042506	020104	046122		
	033752	040504	000			
3321	033755	127	044522	044524	EM75A:	.ASCIZ /WRITING RLBA MODIFED RLBAE/
	033762	043516	051040	041114		
	033770	020101	047515	044504		
	033776	044506	042105	051040		
	034004	041114	042501	000		
3322	034011	127	044522	044524	EM76:	.ASCIZ /WRITING RLDA MODIFED RLCS/
	034016	043516	051040	042114		
	034024	020101	047515	044504		
	034032	044506	042105	051040		
	034040	041514	000123			
3323	034044	051127	052111	047111	EM77:	.ASCIZ /WRITING RLDA MODIFED RLBA/
	034052	020107	046122	040504		
	034060	046440	042117	043111		
	034066	042511	020104	046122		
	034074	040502	000			
3324	034077	127	044522	044524	EM77A:	.ASCIZ /WRITING RLDA MODIFED RLBAE/
	034104	043516	051040	042114		
	034112	020101	047515	044504		
	034120	044506	042105	051040		
	034126	041114	042501	000		
3325	034133	116	046530	042440	EM80:	.ASCII /NXM ERROR FLAG DIDN'T SET/
	034140	051122	051117	043040		

	034146	040514	020107	044504	
	034154	047104	052047	051440	
	034162	052105			
3326	034164	020054	040502	045516	EM80X: .ASCIZ /, BANK SELECT 7 (BBS7) FAILS/
	034172	051440	046105	041505	
	034200	020124	020067	041050	
	034206	051502	024467	043040	
	034214	044501	051514	000	
3327	034221	123	040524	052524	EM90: .ASCIZ /STATUS INCORRECT AFTER NON-EX MEMORY ACCESS/
	034226	020123	047111	047503	
	034234	051122	041505	020124	
	034242	043101	042524	020122	
	034250	047516	026516	054105	
	034256	046440	046505	051117	
	034264	020131	041501	042503	
	034272	051523	000		
3328	034275	125	042516	050130	EM91: .ASCIZ /UNEXPECTED STATUS ERROR ON EXT MEMORY ACCESS/
	034302	041505	042524	020104	
	034310	052123	052101	051525	
	034316	042440	051122	051117	
	034324	047440	020116	054105	
	034332	020124	042515	047515	
	034340	054522	040440	041503	
	034346	051505	000123		
3329	034352	040504	040524	044440	EM92: .ASCIZ \DATA INCORRECT TO/FROM EXTENDED MEMORY\
	034360	041516	051117	042522	
	034366	052103	052040	027517	
	034374	051106	046517	042440	
	034402	052130	047105	042504	
	034410	020104	042515	047515	
	034416	054522	000		
3330	034421	125	042516	050130	EM98: .ASCIZ /UNEXPECTED CONTROLLER ERRORS/
	034426	041505	042524	020104	
	034434	047503	052116	047522	
	034442	046114	051105	042440	
	034450	051122	051117	000123	
3331	034456	000170			EM99: .BLKB 120.
3332					;
3333					; TLM UNIQUE ERROR TEXT.
3334					;
3335	034646	052502	020123	044524	EM100: .ASCIZ /BUS TIME-OUT ON TLM ADDRESS/
	034654	042515	047455	052125	
	034662	047440	020116	046124	
	034670	020115	042101	051104	
	034676	051505	000123		
3336	034702	046124	020115	052123	EM100A: .ASCII /TLM STATUS WRONG/<15><12>
	034710	052101	051525	053440	
	034716	047522	043516	005015	
3337	034724	044103	041505	020113	.ASCIZ /CHECK THAT CABLE IS IN PLACE !!!/
	034732	044124	052101	041440	
	034740	041101	042514	044440	
	034746	020123	047111	050040	
	034754	040514	042503	020440	
	034762	020441	000		
3338	034765	120	047522	020115	EM100B: .ASCII /PROM ID WRONG/<15><12>
	034772	042111	053440	047522	

3339	035000	043516	005015		
	035004	044103	041505	020113	.ASCIZ /CHECK THAT CORRECT PROM SET IS INSTALLED !!!/
	035012	044124	052101	041440	
	035020	051117	042522	052103	
	035026	050040	047522	020115	
	035034	042523	020124	051511	
	035042	044440	051516	040524	
	035050	046114	042105	020440	
	035056	020441	000		
3340	035061	123	051531	046103	EM101: .ASCIZ \SYSCLK<11> AND/OR PWROK<8> NOT SET IN TLM STATUS\
	035066	036113	030461	020076	
	035074	047101	027504	051117	
	035102	050040	051127	045517	
	035110	034074	020076	047516	
	035116	020124	042523	020124	
	035124	047111	052040	046514	
	035132	051440	040524	052524	
	035140	000123			
3341	035142	051104	053111	020105	EM102: .ASCIZ /DRIVE READY STATUS BIT<0> INCORRECT IN RLCSR/
	035150	042522	042101	020131	
	035156	052123	052101	051525	
	035164	041040	052111	030074	
	035172	020076	047111	047503	
	035200	051122	041505	020124	
	035206	047111	051040	041514	
	035214	051123	000		
3342	035217	104	044522	042526	EM103: .ASCIZ /DRIVE ERROR STATUS BITS<15:14> INCORRECT IN RLCSR/
	035224	042440	051122	051117	
	035232	051440	040524	052524	
	035240	020123	044502	051524	
	035246	030474	035065	032061	
	035254	020076	047111	047503	
	035262	051122	041505	020124	
	035270	047111	051040	041514	
	035276	051123	000		
3343	035301	104	044522	042526	EM104: .ASCII /DRIVE /
	035306	040			
3344	035307	060	051440	046105	EM104X: .ASCIZ /0 SELECT BITS<10:9> INCORRECT IN TLM STATUS/
	035314	041505	020124	044502	
	035322	051524	030474	035060	
	035330	037071	044440	041516	
	035336	051117	042522	052103	
	035344	044440	020116	046124	
	035352	020115	052123	052101	
	035360	051525	000		
3345	035363	104	044522	042526	EM105: .ASCIZ /DRIVE COMMAND (GET STATUS) INCORRECT IN TLM CSR+2/
	035370	041440	046517	040515	
	035376	042116	024040	042507	
	035404	020124	052123	052101	
	035412	051525	020051	047111	
	035420	047503	051122	041505	
	035426	020124	047111	052040	
	035434	046514	041440	051123	
	035442	031053	000		
3346	035445	122	052105	051125	EM106: .ASCIZ /RETURNED DRIVE STATUS INCORRECT IN RLMP/
	035452	042516	020104	051104	

	035460	053111	020105	052123	
	035466	052101	051525	044440	
	035474	041516	051117	042522	
	035502	052103	044440	020116	
	035510	046122	050115	000	
3347	035515	105	051122	051117	EM107: .ASCIZ /ERRORS IN RLCSR AFTER "GET STATUS" COMMAND/
	035522	020123	047111	051040	
	035530	041514	051123	040440	
	035536	052106	051105	021040	
	035544	042507	020124	052123	
	035552	052101	051525	020042	
	035560	047503	046515	047101	
	035566	000104			
3348	035570	051104	053111	020105	EM110: .ASCIZ /DRIVE COMMAND (SEEK DIFF) INCORRECT IN TLM CSR*2/
	035576	047503	046515	047101	
	035604	020104	051450	042505	
	035612	020113	044504	043106	
	035620	020051	047111	047503	
	035626	051122	041505	020124	
	035634	047111	052040	046514	
	035642	041440	051123	031053	
	035650	000			
3349	035651	105	051122	051117	EM111: .ASCIZ /ERRORS IN RLCSR AFTER "SEEK" COMMAND/
	035656	020123	047111	051040	
	035664	041514	051123	040440	
	035672	052106	051105	021040	
	035700	042523	045505	020042	
	035706	047503	046515	047101	
	035714	000104			
3350	035716	051127	052111	020105	EM112: .ASCIZ /WRITE GATE<13> OR DATA ACTIVE<12> DIDN'T SET DURING WRITE/
	035724	040507	042524	030474	
	035732	037063	047440	020122	
	035740	040504	040524	040440	
	035746	052103	053111	036105	
	035754	031061	020076	044504	
	035762	047104	052047	051440	
	035770	052105	042040	051125	
	035776	047111	020107	051127	
	036004	052111	000105		
3351	036010	051127	052111	020105	EM113: .ASCIZ /WRITE GATE ERROR BIT<14> SET DURING WRITE/
	036016	040507	042524	042440	
	036024	051122	051117	041040	
	036032	052111	030474	037064	
	036040	051440	052105	042040	
	036046	051125	047111	020107	
	036054	051127	052111	000105	
3352	036062	051127	052111	020105	EM113A: .ASCIZ /WRITE GATE BIT<13> STILL SET AFTER WRITE DONE/
	036070	040507	042524	041040	
	036076	052111	030474	037063	
	036104	051440	044524	046114	
	036112	051440	052105	040440	
	036120	052106	051105	053440	
	036126	044522	042524	042040	
	036134	047117	000105		
3353	036140	051105	047522	051522	EM114: .ASCIZ /ERRORS IN RLCSR AFTER WRITE DATA COMMAND/
	036146	044440	020116	046122	

	036154	051503	020122	043101	
	036162	042524	020122	051127	
	036170	052111	020105	040504	
	036176	040524	041440	046517	
	036204	040515	042116	000	
3354	036211	105	051122	051117	EM115: .ASCII /ERROR STATUS INCORRECT AFTER READ DATA/
	036216	051440	040524	052524	
	036224	020123	047111	047503	
	036232	051122	041505	020124	
	036240	043101	042524	020122	
	036246	042522	042101	042040	
	036254	052101	101		
3355	036257	000	044527	044124	EM115X: .ASCIZ <0>/WITHOUT HEADER CHECK/
	036264	052517	020124	042510	
	036272	042101	051105	041440	
	036300	042510	045503	000	
3356	036305	122	040505	020104	EM116: .ASCIZ /READ HEADER CAN'T FIND TLM SECTOR 2 (CYL1,SEC6)/
	036312	042510	042101	051105	
	036320	041440	047101	052047	
	036326	043040	047111	020104	
	036324	046124	020115	042523	
	036342	052103	051117	031040	
	036350	024040	054503	030514	
	036356	051454	041505	024466	
	036364	000			
3357	036365	122	041505	044505	EM117: .ASCII /RECEIVED DATA INCORRECT AFTER READ/
	036372	042526	020104	040504	
	036400	040524	044440	041516	
	036406	051117	042522	052103	
	036414	040440	052106	051105	
	036422	051040	040505	104	
3358	036427	000	044527	044124	EM117X: .ASCIZ <0>/WITHOUT HEADER CHECK/
	036434	052517	020124	042510	
	036442	042101	051105	041440	
	036450	042510	045503	000	
3359	036455	105	051122	051117	EM118: .ASCIZ /ERROR STATUS INCORRECT AFTER WRITE-CHECK/
	036462	051440	040524	052524	
	036470	020123	047111	047503	
	036476	051122	041505	020124	
	036504	043101	042524	020122	
	036512	051127	052111	026505	
	036520	044103	041505	000113	
3360	036526	051120	046517	051040	EM119: .ASCII /PROM READ FAILURE/<15><12>
	036534	040505	020104	040506	
	036542	046111	051125	006505	
	036550	012			
3361	036551	103	047101	052047	.ASCII /CAN'T FIND SECTOR /
	036556	043040	047111	020104	
	036564	042523	052103	051117	
	036572	040			
3362	036573	061	000		EM119X: .ASCIZ /1/
3363	036575	123	041505	047524	TSEC: .ASCII /SECTOR /
	036602	020122			
3364	036604	026061	000040		TSECX: .ASCIZ /1. /
3365	036610	047516	044515	040516	TCLKO: .ASCIZ /NOMINAL CLOCK/
	036616	020114	046103	041517	

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 46-17
GLOBAL ERROR HANDLERS AND ASCII TEXT

SEQ 0130

3366	036624	000113				
	036626	040506	052123	041440	TCLK1:	.ASCIZ /FAST CLOCK/
	036634	047514	045503	000		
3367	036641	123	047514	020127	TCLK2:	.ASCIZ /SLOW CLOCK/
	036646	046103	041517	000113		
3368	036654	026440	046440	054101	TPEAK:	.ASCIZ / - MAX PEAK SHIFT/
	036662	050040	040505	020113		
	036670	044123	043111	000124		
3369	036676	054105	023520	035104	EXPHCRC:	.ASCIZ /EXP'D: ERR HCRC OPI/
	036704	042440	051122	044040		
	036712	051103	020103	050117		
	036720	000111				
3370	036722	054105	023520	035104	EXPDCRC:	.ASCIZ /EXP'D: ERR DCRC/
	036730	042440	051122	042040		
	036736	051103	000103			
3371	036742	054105	023520	035104	EXPNON:	.ASCIZ /EXP'D: NONE/
	036750	047040	047117	000105		
3372						.EVEN

```

3374 .SBTTL DEVELOPMENT/DEBUG AIDS
3375 .SBTTL
3376 .SBTTL RLV12 EMULATOR
3377 ;
3378 ; FOR DEBUGGING (UNDER MIMIC) WE'LL EMULATE THE ACTION
3379 ; THAT WOULD OCCUR ON A MAINT/NOP FUNCTION.
3380 ;
3381 ;LWL01 .IIF P2, .PRINT . ; FOR DEBUG, MIMIC FLAG IS HERE...
3382 036756 000000 MIMTC: 0 ;...SET NZ TO ENABLE EMULATOR.
3383 ;
3384 036760 023727 003402 000000 EMURLV: CMP RL TYP,#RL11
3385 036766 001004 BNE 1$ ; BR IF RLV
3386 036770 042777 036000 144326 BIC #36000,@RLCS ; RL11 -- NOP -- CLEAR ERROR BITS.
3387 036776 000517 BR 7$ ; AND THAT'S ALL.
3388 ;
3389 037000 027727 144326 177001 1$: CMP @RLMP,#-511. ; WORD COUNT RIGHT ??
3390 037006 001404 BEQ 2$ ; YES.
3391 037010 052777 112000 144306 BIS #ERR!HNF!OPI,@RLCS ; NO SET THE ERROR BITS...
3392 037016 000507 BR 7$ ;...AND RETURN.
3393 ;
3394 037020 020527 020470 2$: CMP R5,#BB7 ; CALLED FROM BBS7 TEST ??
3395 037024 001004 BNE 3$ ; SKIP IF NOT.
3396 037026 052777 120000 144270 BIS #ERR!NXM,@RLCS ; YES, SET ERROR BITS...
3397 037034 000500 BR 7$ ;...AND RETURN.
3398 ;
3399 037036 012701 004366 3$: MOV #BUF1,R1 ; STANDARD SRC...
3400 037042 012702 005366 MOV #BUF2,R2 ;...AND DST...
3401 037046 012703 037310 MOV #DFIFO,R3 ;...AND DUMMY FIFO POINTERS.
3402 037052 023701 010416 CMP BA16,R1 ; IS BA STANDARD (INTERNAL) ??
3403 037056 001411 BEQ 4$ ; BR IF SO.
3404 037060 012701 100000 MOV #100000,R1 ; POINT TO BUF1 IF EXT MEMORY.
3405 037064 012702 101000 MOV #101000,R2 ; DITTO BUF2 (RECEIVER).
3406 037070 012737 037160 000004 MOV #5$,ERRVEC ; SET TRAP CATCHER.
3407 037076 005237 177572 INC MMRO ;***** KT ON *****
3408 037102 062777 000002 144216 4$: ADD #2,@RLBA ; SIMULATE INC BA... *
3409 037110 005577 144220 ADC @RLBAE ;...OVERFLOWING INTO BAE. *
3410 037114 012123 MOV (R1)+,(R3)+ ; MEM => FIFO (256 WORDS). *
3411 037116 000240 240 ;
3412 037120 020327 040310 CMP R3,#DFIFO+1000 ; *
3413 037124 103766 BLO 4$ ; *
3414 037126 012703 037310 MOV #DFIFO,R3 ; RESET FIFO POINTER. *
3415 037132 062777 000002 144166 14$: ADD #2,@RLBA ; *
3416 037140 005577 144170 ADC @RLBAE ; INC BA. *
3417 037144 012322 MOV (R3)+,(R2)+ ; FIFO => MEM (255 WORDS). *
3418 037146 000240 240 ;
3419 037150 020327 040306 CMP R3,#DFIFO+776 ; *
3420 037154 103766 BLO 14$ ; LOOP... *
3421 037156 000406 BR 6$ ;...UNTIL DONE. *
3422 037160 052777 120000 144136 5$: BIS #ERR!NXM,@RLCS ; ON NXM TRAP, SET ERROR BITS... *
3423 037166 012716 037174 MOV #6$,(SP) ;...AND CONTINUE. *
3424 037172 000002 RTI ; *
3425 037174 005037 177572 6$: CLR MMRO ;***** KT OFF *****
3426 037200 012737 010670 000004 MOV #TRAP4,ERRVEC ; RESET TRAP CATCHER.
3427 037206 017700 144116 MOV @RLDA,RO ;
3428 037212 062700 000006 ADD #6,RO ; ADJUST FINAL DA...
3429 037216 110077 144106 MOVB RO,@RLDA ;...LO BYTE ONLY.

```

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 47-1
RLV12 EMULATOR

SEQ 0132

3430	037222	013777	003442	144102		MOV	GDCRC3,@RLMP		; FAKE THE 1ST...
3431	037230	013737	003444	003374		MOV	GDCRC4,E.MP1		; ...AND 2ND CRC WORDS.
3432	037236	052777	000200	144060	7\$:	BIS	#CRDY,@RLCS		; SET DONE BIT...
3433	037244	032777	000100	144052		BIT	#INTEN,@RLCS		; INTERRUPT EXPECTED ??
3434	037252	001410				BEQ	8\$; NO.
3435	037254					GETPRI	RO		
(3)	037254	104440							TRAP C\$GPRI
3436	037256	042700	177437			BIC	#+CPRI07,RO		; MASK PRIORITY BITS.
3437	037262	020037	003340			CMP	RO,BPRIOR		
3438	037266	002002				BGE	8\$; BR IF CPU >= RL BR LEVEL
3439	037270	005237	010700			INC	INTFLG		; OTHERWISE, SET INT RECEIVED FLAG.
3440	037274	000207			8\$:	RTS	PC		; ...AND RETURN.
3441									
3442	037276	000005				DUMMY:	.BLKW 5		; THESE ARE THE DUMMY REGISTERS.
3443	037310	000400				DFIFO:	.BLKW 256.		; AND A DUMMY FIFO.

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33MACY11 30(1046) 12-JUN-85 15:42 PAGE 48-4
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0138

EM14B	031554	1827	3284#				
EM15	031616	1852	2183	3285#			
EM16	031660	1885	3286#				
EM17	031713	1949	3287#				
EM2	031005	1188	3271#				
EM20	031760	2081	3288#				
EM21	032031	2087	3289#				
EM22	032111	2101	2197	3290#			
EM23	032155	2117	3291#				
EM27	032223	1921	3292#				
EM27X	032323	1910*	1911*	1926*	1927*	3294#	
EM3	031032	1210	3272#				
EM31	032332	1980	3295#				
EM32	032360	1987	3296#				
EM4	031057	1232	3273#				
EM4A	031104	1257	3274#				
EM4B	031132	1263	3275#				
EM44	032405	1718	3297#				
EM45	032440	1724	3298#				
EM46	032473	1730	3299#				
EM47	032526	1738	3300#				
EM5	031201	1336	3276#				
EM50	032562	1765	3301#				
EM51	032621	1770	3302#				
EM52	032645	1774	3303#				
EM6	031252	1363	3277#				
EM61	032671	1443	3304#				
EM62	032737	1456	3305#				
EM63	033007	1484	3306#				
EM64	033055	1493	3307#				
EM65	033125	1517	3308#				
EM66	033173	1526	3309#				
EM67	033243	1554	3310#				
EM68	033312	1564	3311#				
EM69	033363	1300	3312#				
EM7	031300	1386	3278#				
EM70	033417	1304	3313#				
EM71	033453	1307	3314#				
EM71A	033507	1312	3315#				
EM72	033544	1594	3316#				
EM73	033577	1600	3317#				
EM73A	033632	1608	3318#				
EM74	033671	1635	3319#				
EM75	033723	1641	3320#				
EM75A	033755	1649	3321#				
EM76	034011	1676	3322#				
EM77	034044	1682	3323#				
EM77A	034077	1690	3324#				
EM8	031326	1413	3279#				
EM80	034133	2255	3325#				
EM80X	034164	2233*	2235*	3326#			
EM90	034221	2336	3327#				
EM91	034275	2341	3328#				
EM92	034352	2369	3329#				
EM98	034421	2056	2177	3330#			
EM99	034456	671	3144	3150	3157	3180	3331#

Symbol	Value	276#	493*	524	580*	886	893	899*
LONUM	003454	276#	493*	524	580*	886	893	899*
LOT	000010 G	155#						
LSI	006576	503	513#					
L#ACP	002110 G	50#						
L#APT	002036 G	50#						
L#AU	007476 G	50	648#					
L#AUT	002070 G	50#						
L#AUTO	007260 G	50	605#					
L#CCP	002106 G	50#						
L#CLEA	007336 G	50	620#					
L#CO	002032 G	50#						
L#DEPO	002011 G	50#						
L#DESC	002122 G	50	52#					
L#DESP	002076 G	50#						
L#DEVP	002060 G	50#						
L#DISP	026124 G	50	3118*	3121#				
L#DLY	002116 G	50#						
L#DTP	002040 G	50#						
L#DTYP	002034 G	50#						
L#DU	007414 G	50	636#					
L#DUT	002072 G	50#						
L#DVTY	002142 G	50	54#					
L#EF	002052 G	50#						
L#ENVI	002044 G	50#						
L#ETP	002102 G	50#						
L#EXP1	002046 G	50#						
L#EXP4	002064 G	50#						
L#EXP5	002066 G	50#						
L#HARD	002172 G	50	63#					
L#HIME	002120 G	50#						
L#HPCP	002016 G	50#						
L#HPTP	002022 G	50#						
L#HW	002376 G	50	89#					
L#ICP	002104 G	50#						
L#INIT	006376 G	50	473#					
L#LADP	002026 G	50#						
L#LAST	040416 G	45	50	3453#	3468			
L#LOAD	002100 G	50#						
L#LUN	002074 G	50#						
L#MREV	002050 G	50#						
L#NAME	002000 G	50#						
L#PRIO	002042 G	50#						
L#PROT	006370 G	50	465#					
L#PRT	002112 G	50#						
L#REPP	002062 G	50#						
L#REV	002010 G	50#						
L#RPT	011722 G	50	1130#					
L#SOFT	002412 G	50	105#					
L#SPC	002056 G	50#						
L#SPCP	002020 G	50#						
L#SPTP	002024 G	50#						
L#STA	002030 G	50#						
L#SW	003304 G	50	143#					
L#TEST	002114 G	50#						
L#TIML	002014 G	50#						
L#UNIT	002012 G	50#	520	529	606	637		

CVRLBBO -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 48-10
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0144

L10000	002266	63	74#						
L10001	002410	89	99#						
L10002	002542	105	120#						
L10003	003322	143	151#						
L10005	007256	601#							
L10006	007334	616#							
L10007	007412	632#							
L10010	007474	644#							
L10011	007552	656#							
L10012	012170	1138	1144#						
L10013	012254	1169#							
L10014	012340	1191#							
L10015	012424	1213#							
L10016	012510	1235#							
L10017	012632	1264#							
L10020	013074	1313#							
L10021	013214	1342#							
L10022	013320	1369#							
L10023	013406	1392#							
L10024	013512	1419#							
L10025	013740	1462#							
L10026	014124	1499#							
L10027	014272	1532#							
L10030	014464	1570#							
L10031	014666	1609#							
L10032	015112	1650#							
L10033	015336	1691#							
L10034	015626	1739#							
L10035	016022	1775#							
L10036	016230	1818	1829#						
L10037	016316	1854#							
L10040	016464	1892#							
L10041	016620	1929#							
L10042	016704	1952#							
L10043	017032	1988#							
L10044	017732	2046	2121#						
L10045	017100	2020#							
L10046	017146	2035#							
L10047	017176	2045#							
L10050	020356	2160	2213#						
L10051	017764	2146#							
L10052	020020	2159#							
L10053	020552	2232	2259#						
L10054	021346	2284	2384#						
L10055	021720	2451#							
L10056	022124	2488#							
L10057	022272	2519#							
L10060	022552	2566#							
L10061	023434	2626	2695#						
L10062	024032	2751	2771#						
L10063	027436	3222#							
L10064	040422	3461#							
L10066	040434	3461	3467#						
MAINT =	000000	181#	1913	1945	1974	2052	2173	2245	2315
MFPT =	000007	508#	509						
MIMIC	036756	534	755	816	3382#				

		1462#	1474#	1499#	1510#	1532#	1546#	1570#	1609#	1650#	1691#	1739#	1775#	1829#
		1854#	1870#	1892#	1929#	1952#	1988#	2020#	2035#	2045#	2050#	2121#	2146#	2159#
		2164#	2213#	2259#	2296#	2384#	2451#	2488#	2519#	2566#	2632#	2695#	2755#	2771#
		3222#												
TAB	026030	2997	3095#											
TCLK0	036610	2594	2719	3365#										
TCLK1	036626	2599	2724	3366#										
TCLK2	036641	2604	2729	3367#										
TCSR	003344	240#	542*	544*	547*	2401*	2402*	2403	2415*	2423*	2431*	2437	2781	2784*
		2789	2813	3111										
TCSU0	002776	110	130#											
TCS0	003320	150#	544	2940										
TEMHI	003460	278#	525*	581										
TEML0	003456	277#	524*	580	2041									
TEMP1	003422	263#	946*	947	949*	950	957*	958						
TEMP2	003412	259#	984*	1007*										
TEMP3	003414	260#	985*	989*										
TEMP4	003416	261#	986*	988	998*	1000	1005*	1006*	1009					
TEMP5	003420	262#	947*	948*	949	956*	957							
TLMBP	002575	107	126#											
TLMF	003316	149#	542											
TLMIN	002751	108	129#											
TLMOK	024034	2399	2460	2496	2527	2778#								
TLMPID	024040	2578	2703	2780#										
TLM1	021350	2398#												
TLM2	021722	2459#												
TLM3	022126	2495#												
TLM4	022274	2526#												
TLM5	022554	2577#												
TLM6	023436	2702#												
TMPO	003424	264#	836*	839	842*	1851*	1868*	1888*	2069*	2070*	2071*	2072	2097*	2103
		2193*	2199	2358*	2684*	2981*	2984	3147	3164	3188				
TMP1	003426	265#	2098*	2103	2194*	2199	2253*	2327*	2361*	2363*	2368*	2486*	2517*	2564*
		2650*	2674*	2685*	2767*	3172*	3173	3180	3188					
TMP2	003430	266#	2089*	2099	2102*	2110	2112	2185*	2195	2198*	2206	2208	2254*	2328*
		2360*	2362*	2367*	2594*	2599*	2604*	2719*	2724*	2729*	3169	3171*	3173	3192
TPEAK	036654	2608*	2613*	2733*	2738*	3192	3368#							
TRAP4	010670	597	908#	2311	3426									
TRPFLG	010672	608*	611	909#	1158*	1163	1180*	1185	1202*	1207	1224*	1229	1246*	1255
		1261	2279*	2282	2301*	2307*	2329	2783*	2786					
TSEC	036575	3192	3363#											
TSECX	036604	2588	2713	2852*	2856*	2862*	2868*	2872*	2880*	3364#				
TURKY	024562	2876	2892#											
T\$ARGC=	000001	50#	640#	652#	1135#	1136#	1137#	2103#	2112#	2199#	2208#	2876#	3043#	3082#
		3132#	3135#	3138#	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#
		3188#	3189#	3192#	3204#	3205#	3206#	3207#	3219#					
T\$CODE=	004032	64#	65#	66#	67#	68#	69#	70#	71#	72#	73#	106#	107#	108#
		109#	110#	111#	112#	113#	114#	115#	116#	117#	118#	119#		
T\$ERRN=	000264	17#	614#	929#	1166#	1188#	1210#	1232#	1257#	1263#	1300#	1304#	1307#	1312#
		1336#	1363#	1386#	1413#	1443#	1456#	1484#	1493#	1517#	1526#	1554#	1564#	1594#
		1600#	1608#	1635#	1641#	1649#	1676#	1682#	1690#	1718#	1724#	1730#	1738#	1765#
		1770#	1774#	1808#	1816#	1827#	1852#	1885#	1921#	1949#	1980#	1987#	2056#	2065#
		2075#	2081#	2087#	2101#	2117#	2177#	2183#	2197#	2255#	2336#	2341#	2369#	2407#
		2414#	2422#	2430#	2446#	2478#	2483#	2487#	2514#	2518#	2549#	2559#	2561#	2565#
		2589#	2651#	2665#	2675#	2686#	2714#	2768#	2800#	2802#	2804#			
T\$EXCP=	000000	69#	70#	71#	73#	106#	110#	113#	119#					

T\$FLAG= 000040	1138#	1337#	1364#	1387#	1414#	1444#	1457#	1485#	1494#	1518#	1527#	1555#	1565#
T\$FREE= 040434	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#					
T\$GMAN= 000000	3453	3468#											
T\$HILI= 000006	17#												
T\$LAST= 000001	69#	70#	71#	73#	106#	110#	113#	119#					
T\$LOLI= 000000	17#	3453#	3460										
T\$LSYM= 010000	69#	70#	71#	73#	106#	110#	113#	119#					
	17#	74	99	120	151	601	616	632	644	656	1144	1169	1191
	1213	1235	1264	1313	1342	1369	1392	1419	1462	1499	1532	1570	1609
	1650	1691	1739	1775	1829	1854	1892	1929	1952	1988	2020	2035	2045
	2121	2146	2159	2213	2259	2384	2451	2488	2519	2566	2695	2771	3222
T\$LTNO= 000043	3453#												
T\$NEST= 177777	17#	63#	65	67	72	74#	89#	99#	105#	109	112	120#	143#
	151#	465#	469#	473#	601#	605#	616#	620#	632#	636#	644#	648#	656#
	1130#	1144#	1157#	1169#	1179#	1191#	1201#	1213#	1223#	1235#	1243#	1264#	1279#
	1313#	1323#	1325#	1341#	1342#	1352#	1354#	1368#	1369#	1378#	1380#	1391#	1392#
	1402#	1406#	1418#	1419#	1429#	1431#	1461#	1462#	1472#	1474#	1498#	1499#	1508#
	1510#	1531#	1532#	1542#	1546#	1569#	1570#	1583#	1609#	1620#	1650#	1661#	1691#
	1702#	1739#	1749#	1775#	1786#	1829#	1841#	1854#	1866#	1870#	1890#	1892#	1902#
	1929#	1939#	1952#	1967#	1988#	2005#	2009#	2020#	2024#	2035#	2039#	2045#	2050#
	2118#	2121#	2133#	2137#	2146#	2150#	2159#	2164#	2209#	2213#	2227#	2259#	2278#
	2296#	2372#	2384#	2398#	2451#	2459#	2488#	2495#	2519#	2526#	2566#	2577#	2632#
	2693#	2695#	2702#	2755#	2769#	2771#	3127#	3222#					
T\$NSO = 000011	63#	65	67	72	74	89#	99	105#	109	112	120	143#	151
	465#	469	473#	601	605#	616	620#	632	636#	644	648#	656	1130#
	1144	1157#	1169	1179#	1191	1201#	1213	1223#	1235	1243#	1264	1279#	1313
	1323#	1342	1352#	1369	1378#	1392	1402#	1419	1429#	1462	1472#	1499	1508#
	1532	1542#	1570	1583#	1609	1620#	1650	1661#	1691	1702#	1739	1749#	1775
	1786#	1829	1841#	1854	1866#	1892	1902#	1929	1939#	1952	1967#	1988	2005#
	2121	2133#	2213	2227#	2259	2278#	2384	2398#	2451	2459#	2488	2495#	2519
	2526#	2566	2577#	2695	2702#	2771	3127#	3222					
T\$NS1 = 000003	1325#	1341	1354#	1368	1380#	1391	1406#	1418	1431#	1461	1474#	1498	1510#
	1531	1546#	1569	1870#	1890	2009#	2020	2024#	2035	2039#	2045	2050#	2118
	2137#	2146	2150#	2159	2164#	2209	2296#	2372	2632#	2693	2755#	2769	
T\$PCNT= 000000	3460#	3461#											
T\$PTAB= 010065	3461#												
T\$PTHV= 000001	50	3468#											
T\$PTNU= 000001	17#	3461#	3468										
T\$SAVL= 177777	17#												
T\$SEGL= 177777	17#	1325#	1337	1341#	1354#	1364	1368#	1380#	1387	1391#	1406#	1414	1418#
	1431#	1444	1457	1461#	1474#	1485	1494	1498#	1510#	1518	1527	1531#	1546#
	1555	1565	1569#	1870#	1890#	2050#	2060	2118#	2164#	2209#	2296#	2372#	2632#
	2693#	2755#	2769#										
T\$SEKO= 010000	1325#	1337	1341	1354#	1364	1368	1380#	1387	1391	1406#	1414	1418	1431#
	1444	1457	1461	1474#	1485	1494	1498	1510#	1518	1527	1531	1546#	1555
	1565	1569	1870#	1890	2050#	2060	2118	2164#	2209	2296#	2372	2632#	2693
	2755#	2769											
T\$SIZE= 000007	3453	3468#											
T\$SUBN= 000000	17#	1157#	1179#	1201#	1223#	1243#	1279#	1323#	1352#	1378#	1402#	1429#	1472#
	1508#	1542#	1583#	1620#	1661#	1702#	1749#	1786#	1841#	1866#	1902#	1939#	1967#
	2005#	2009#	2024#	2039#	2133#	2137#	2150#	2227#	2278#	2398#	2459#	2495#	2526#
	2577#	2702#											
T\$TAGL= 177777	17#												
T\$TAGN= 010067	17#	63#	89#	105#	143#	465#	473#	605#	620#	636#	648#	1130#	1157#
	1179#	1201#	1223#	1243#	1279#	1323#	1352#	1378#	1402#	1429#	1472#	1508#	1542#
	1583#	1620#	1661#	1702#	1749#	1786#	1841#	1866#	1902#	1939#	1967#	2005#	2009#

BCOMPL	475	477	479	481	483	3213											
BEGIN.	219#	1157	1179	1201	1223	1243	1279	1323	1352	1378	1402	1429	1472	1508	1542		
	1583	1620	1661	1702	1749	1786	1841	1866	1902	1939	1967	2005	2133	2227	2278		
	2398	2459	2495	2526	2577	2702											
BGNAU	648																
BGNAUT	605																
BGNCLN	620																
BGNDU	636																
BGNHRD	63																
BGNHW	89																
BGNINI	473																
BGNMSG	3127																
BGNPRO	465																
BGNPTA	3461																
BGNRPT	1130																
BGNSEG	1325	1354	1380	1406	1431	1474	1510	1546	1870	2050	2164	2296	2632	2755			
BGNSET	3460																
BGNSFT	105																
BGNSUB	2009	2024	2039	2137	2150												
BGNSW	143																
BGNTST	1157	1179	1201	1223	1243	1279	1323	1352	1378	1402	1429	1472	1508	1542	1583		
	1620	1661	1702	1749	1786	1841	1866	1902	1939	1967	2005	2133	2227	2278	2398		
	2459	2495	2526	2577	2702												
BNCOMP	532																
CKLOOP	1595	1601	1636	1642	1677	1683	1719	1725	1731	1766	1771	1809	1817	1828	1836		
	1922	1950	2057	2066	2076	2082	2088	2104	2178	2184	2200	2371					
DESCRI	52																
DEVTYP	54																
DFERR	212#	929	1263	1300	1304	1307	1312	1336	1363	1386	1413	1443	1456	1484	1493		
	1517	1526	1554	1564	1594	1600	1608	1635	1641	1649	1676	1682	1690	1718	1724		
	1730	1738	1765	1770	1774	1808	1816	1827	1852	1885	1921	1949	1980	1987	2056		
	2065	2075	2081	2087	2101	2117	2177	2183	2197	2255	2336	2341	2369	2407	2414		
	2422	2430	2446	2478	2483	2487	2514	2518	2549	2559	2561	2565	2589	2651	2665		
	2675	2686	2714	2768	2800	2802	2804										
DISPAT	3121																
DISPLA	107																
DOCLN	522	1168	1190	1212	1234	1259	3080	3221									
DODU	615	1167	1189	1211	1233	1258	3220										
DORPT	518																
ENDAU	656																
ENDAUT	616																
ENDCLN	632																
ENDDU	644																
ENDHRD	74																
ENDHW	99																
ENDINI	601																
ENDMSG	3222																
ENDPRO	469																
ENDPPTA	3467																
ENDRPT	1144																
ENDSEG	1341	1368	1391	1418	1461	1498	1531	1569	1890	2118	2209	2372	2693	2769			
ENDSET	3468																
ENDSFT	120																
ENDSUB	2020	2035	2045	2146	2159												
ENDSW	151																
ENDTST	1169	1191	1213	1235	1264	1313	1342	1369	1392	1419	1462	1499	1532	1570	1609		

	1650	1691	1739	1775	1829	1854	1892	1929	1952	1988	2121	2213	2259	2384	2451
EQUALS	2488	2519	2566	2695	2771										
ERRDF	155														
	929	1263	1300	1304	1307	1312	1336	1363	1386	1413	1443	1456	1484	1493	1517
	1526	1554	1564	1594	1600	1608	1635	1641	1649	1676	1682	1690	1718	1724	1730
	1738	1765	1770	1774	1808	1816	1827	1852	1885	1921	1949	1980	1987	2056	2065
	2075	2081	2087	2101	2117	2177	2183	2197	2255	2336	2341	2369	2407	2414	2422
	2430	2446	2478	2483	2487	2514	2518	2549	2559	2561	2565	2589	2651	2665	2675
ERRSF	2686	2714	2768	2800	2802	2804									
ESCAPE	614	1166	1188	1210	1232	1257									
EXIT	1337	1364	1387	1414	1444	1457	1485	1494	1518	1527	1555	1565			
GETPRI	1138	1818	2046	2060	2160	2232	2284	2626	2751						
GPHARD	3435														
GPRMA	531														
GPRMD	69	70	110												
GPRML	71	73	106	113	119										
HEADER	64	66	68	108	111	114	115	116	117	118					
INLOOP	50														
LASTAD	3212														
M\$BYTE	3453														
M\$CHEC	50#														
M\$CNTO	1138#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#						
	64#	66#	68#	69#	70#	71#	73#	106#	108#	110#	111#	113#	114#	115#	116#
M\$COUN	117#	118#	119#												
	640#	652#	1135#	1136#	1137#	2103#	2112#	2199#	2208#	2876#	3043#	3082#	3132#	3135#	3138#
	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#	3188#	3189#	3192#	3204#	3205#
	3206#	3207#	3219#												
M\$DATA	50#	52#	54#												
M\$DECR	74#	99#	120#	151#	469#	601#	616#	632#	644#	656#	1144#	1169#	1191#	1213#	1235#
	1264#	1313#	1341#	1342#	1368#	1369#	1391#	1392#	1418#	1419#	1461#	1462#	1498#	1499#	1531#
	1532#	1569#	1570#	1609#	1650#	1691#	1739#	1775#	1829#	1854#	1890#	1892#	1929#	1952#	1988#
	2020#	2035#	2045#	2118#	2121#	2146#	2159#	2209#	2213#	2259#	2372#	2384#	2451#	2488#	2519#
	2566#	2693#	2695#	2769#	2771#	3222#	3461#								
M\$DEFA	64#	66#	68#	69#	70#	71#	73#	106#	108#	110#	111#	113#	114#	115#	116#
	117#	118#	119#												
M\$ENDE	74#	99#	120#	151#	601#	616#	632#	644#	656#	1144#	1169#	1191#	1213#	1235#	1264#
	1313#	1341#	1342#	1368#	1369#	1391#	1392#	1418#	1419#	1461#	1462#	1498#	1499#	1531#	1532#
	1569#	1570#	1609#	1650#	1691#	1739#	1775#	1829#	1854#	1890#	1892#	1929#	1952#	1988#	2020#
	2035#	2045#	2118#	2121#	2146#	2159#	2209#	2213#	2259#	2372#	2384#	2451#	2488#	2519#	2566#
	2693#	2695#	2769#	2771#	3222#										
M\$ERRI	614#	929#	1166#	1188#	1210#	1232#	1257#	1263#	1300#	1304#	1307#	1312#	1336#	1363#	1386#
	1413#	1443#	1456#	1484#	1493#	1517#	1526#	1554#	1564#	1594#	1600#	1608#	1635#	1641#	1649#
	1676#	1682#	1690#	1718#	1724#	1730#	1738#	1765#	1770#	1774#	1808#	1816#	1827#	1852#	1885#
	1921#	1949#	1980#	1987#	2056#	2065#	2075#	2081#	2087#	2101#	2117#	2177#	2183#	2197#	2255#
	2336#	2341#	2369#	2407#	2414#	2422#	2430#	2446#	2478#	2483#	2487#	2514#	2518#	2549#	2559#
	2561#	2565#	2589#	2651#	2665#	2675#	2686#	2714#	2768#	2800#	2802#	2804#			
M\$ESCA	1337#	1364#	1387#	1414#	1444#	1457#	1485#	1494#	1518#	1527#	1555#	1565#			
M\$ESCS	1337#	1364#	1387#	1414#	1444#	1457#	1485#	1494#	1518#	1527#	1555#	1565#			
M\$EXCP	69#	70#	71#	73#	106#	110#	113#	119#							
M\$EXIT	1138#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#						
M\$EXSE	1138#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#						
M\$EXTJ	1138#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#						
M\$GEN	50#	52#	54#	63#	74#	89#	99#	105#	120#	143#	151#	465#	473#	601#	605#
	616#	620#	632#	636#	644#	648#	656#	1130#	1144#	1157#	1169#	1179#	1191#	1201#	1213#
	1223#	1235#	1243#	1264#	1279#	1313#	1323#	1341#	1342#	1352#	1368#	1369#	1378#	1391#	1392#
	1402#	1418#	1419#	1429#	1461#	1462#	1472#	1498#	1499#	1508#	1531#	1532#	1542#	1569#	1570#

	1583#	1609#	1620#	1650#	1661#	1691#	1702#	1739#	1749#	1775#	1786#	1829#	1841#	1854#	1866#
	1890#	1892#	1902#	1929#	1939#	1952#	1967#	1988#	2005#	2009#	2020#	2024#	2035#	2039#	2045#
	2118#	2121#	2133#	2137#	2146#	2150#	2159#	2209#	2213#	2227#	2259#	2278#	2372#	2384#	2398#
	2451#	2459#	2488#	2495#	2519#	2526#	2566#	2577#	2693#	2695#	2702#	2769#	2771#	3121#	3127#
	3222#	3453#	3461#	3467#											
M#GETS	65#	67#	72#	74#	99#	109#	112#	120#	151#	469#	601#	616#	632#	644#	656#
	1144#	1169#	1191#	1213#	1235#	1264#	1313#	1337#	1341#	1342#	1364#	1368#	1369#	1387#	1391#
	1392#	1414#	1418#	1419#	1444#	1457#	1461#	1462#	1485#	1494#	1498#	1499#	1518#	1527#	1531#
	1532#	1555#	1565#	1569#	1570#	1609#	1650#	1691#	1739#	1775#	1829#	1854#	1890#	1892#	1929#
	1952#	1988#	2020#	2035#	2045#	2060#	2118#	2121#	2146#	2159#	2209#	2213#	2259#	2372#	2384#
M#GETT	2451#	2488#	2519#	2566#	2693#	2695#	2769#	2771#	3222#						
	65#	67#	72#	109#	112#	1138#	1337#	1364#	1387#	1414#	1444#	1457#	1485#	1494#	1518#
M#GNGB	1527#	1555#	1565#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#				
	50#	52#	54#	63#	89#	105#	143#	465#	473#	605#	620#	636#	648#	1130#	3121#
M#GNIN	3127#	3453#													
	50#	52#	54#	63#	64#	65#	66#	67#	68#	69#	70#	71#	72#	73#	74#
	89#	105#	106#	107#	108#	109#	110#	111#	112#	113#	114#	115#	116#	117#	118#
	119#	120#	143#	474#	475#	476#	477#	478#	479#	480#	481#	482#	483#	518#	522#
	531#	532#	601#	614#	615#	616#	631#	632#	640#	644#	652#	656#	929#	1135#	1136#
	1137#	1138#	1144#	1166#	1167#	1168#	1169#	1188#	1189#	1190#	1191#	1210#	1211#	1212#	1213#
	1232#	1233#	1234#	1235#	1257#	1258#	1259#	1263#	1264#	1280#	1300#	1304#	1307#	1312#	1313#
	1325#	1336#	1337#	1341#	1342#	1354#	1363#	1364#	1368#	1369#	1380#	1386#	1387#	1391#	1392#
	1406#	1413#	1414#	1418#	1419#	1431#	1443#	1444#	1456#	1457#	1461#	1462#	1474#	1484#	1485#
	1493#	1494#	1498#	1499#	1510#	1517#	1518#	1526#	1527#	1531#	1532#	1546#	1554#	1555#	1564#
	1565#	1569#	1570#	1594#	1595#	1600#	1601#	1608#	1609#	1635#	1636#	1641#	1642#	1649#	1650#
	1676#	1677#	1682#	1683#	1690#	1691#	1718#	1719#	1724#	1725#	1730#	1731#	1738#	1739#	1765#
	1766#	1770#	1771#	1774#	1775#	1808#	1809#	1816#	1817#	1818#	1827#	1828#	1829#	1843#	1852#
	1853#	1854#	1870#	1871#	1885#	1886#	1890#	1891#	1892#	1921#	1922#	1929#	1942#	1949#	1950#
	1951#	1952#	1980#	1987#	1988#	2009#	2020#	2024#	2035#	2039#	2045#	2046#	2050#	2056#	2057#
	2060#	2065#	2066#	2075#	2076#	2081#	2082#	2087#	2088#	2101#	2103#	2104#	2112#	2117#	2118#
	2121#	2137#	2146#	2150#	2159#	2160#	2164#	2171#	2177#	2178#	2183#	2184#	2184#	2197#	2200#
	2208#	2209#	2210#	2213#	2232#	2255#	2259#	2284#	2296#	2336#	2341#	2369#	2371#	2372#	2384#
	2407#	2414#	2422#	2430#	2446#	2451#	2478#	2483#	2487#	2488#	2514#	2518#	2519#	2549#	2559#
	2561#	2565#	2566#	2589#	2626#	2632#	2651#	2665#	2675#	2686#	2693#	2695#	2714#	2751#	2755#
	2768#	2769#	2771#	2800#	2802#	2804#	2876#	3043#	3080#	3082#	3121#	3132#	3135#	3138#	3141#
	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#	3188#	3189#	3192#	3204#	3205#	3206#
M#GNLS	3207#	3212#	3213#	3219#	3220#	3221#	3222#	3435#	3453#	3461#					
M#GNSU	1341#	1368#	1391#	1418#	1461#	1498#	1531#	1569#	1890#	2118#	2209#	2372#	2693#	2769#	
M#GNTA	2009#	2024#	2039#	2137#	2150#										
	74#	99#	120#	151#	601#	616#	632#	644#	656#	1144#	1169#	1191#	1213#	1235#	1264#
	1313#	1342#	1369#	1392#	1419#	1462#	1499#	1532#	1570#	1609#	1650#	1691#	1739#	1775#	1829#
	1854#	1892#	1929#	1952#	1988#	2020#	2035#	2045#	2121#	2146#	2159#	2213#	2259#	2384#	2451#
M#GNTE	2488#	2519#	2566#	2695#	2771#	3222#	3461#	3467#							
	1157#	1179#	1201#	1223#	1243#	1279#	1323#	1352#	1378#	1402#	1429#	1472#	1508#	1542#	1583#
	1620#	1661#	1702#	1749#	1786#	1841#	1866#	1902#	1939#	1967#	2005#	2133#	2227#	2278#	2398#
	2459#	2495#	2526#	2577#	2702#										
M#HAPT	50#														
M#HNAP	50#														
M#INCR	63#	89#	105#	143#	465#	473#	474#	476#	478#	480#	482#	518#	522#	531#	601#
	605#	614#	615#	616#	620#	631#	632#	636#	640#	644#	648#	652#	656#	929#	1130#
	1135#	1136#	1137#	1144#	1157#	1166#	1167#	1168#	1169#	1179#	1188#	1189#	1190#	1191#	1201#
	1210#	1211#	1212#	1213#	1223#	1232#	1233#	1234#	1235#	1243#	1257#	1258#	1259#	1263#	1264#
	1279#	1280#	1300#	1304#	1307#	1312#	1313#	1323#	1325#	1336#	1337#	1341#	1342#	1352#	1354#
	1363#	1364#	1368#	1369#	1378#	1380#	1386#	1387#	1391#	1392#	1402#	1406#	1413#	1414#	1418#
	1419#	1429#	1431#	1443#	1444#	1456#	1457#	1461#	1462#	1472#	1474#	1484#	1485#	1493#	1494#
	1498#	1499#	1508#	1510#	1517#	1518#	1526#	1527#	1531#	1532#	1542#	1546#	1554#	1555#	1564#

	1565#	1569#	1570#	1583#	1594#	1595#	1600#	1601#	1608#	1609#	1620#	1635#	1636#	1641#	1642#
	1649#	1650#	1661#	1676#	1677#	1682#	1683#	1690#	1691#	1702#	1718#	1719#	1724#	1725#	1730#
	1731#	1738#	1739#	1749#	1765#	1766#	1770#	1771#	1774#	1775#	1786#	1808#	1809#	1816#	1817#
	1818#	1827#	1828#	1829#	1841#	1843#	1852#	1853#	1854#	1866#	1870#	1871#	1885#	1886#	1890#
	1891#	1892#	1902#	1921#	1922#	1929#	1939#	1942#	1949#	1950#	1951#	1952#	1967#	1980#	1987#
	1988#	2005#	2009#	2020#	2024#	2035#	2039#	2045#	2046#	2050#	2056#	2057#	2060#	2065#	2066#
	2075#	2076#	2081#	2082#	2087#	2088#	2101#	2103#	2104#	2112#	2117#	2118#	2121#	2133#	2137#
	2146#	2150#	2159#	2160#	2164#	2171#	2177#	2178#	2183#	2184#	2197#	2199#	2200#	2208#	2209#
	2210#	2213#	2227#	2232#	2255#	2259#	2278#	2284#	2296#	2336#	2341#	2369#	2371#	2372#	2384#
	2398#	2407#	2414#	2422#	2430#	2446#	2451#	2459#	2478#	2483#	2487#	2488#	2495#	2514#	2518#
	2519#	2526#	2549#	2559#	2561#	2565#	2566#	2577#	2589#	2626#	2632#	2651#	2665#	2675#	2686#
	2693#	2695#	2702#	2714#	2751#	2755#	2768#	2769#	2771#	2800#	2802#	2804#	2876#	3043#	3080#
	3082#	3127#	3132#	3135#	3138#	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#
	3188#	3189#	3192#	3204#	3205#	3206#	3207#	3212#	3219#	3220#	3221#	3222#	3435#	3460#	3461#
M#LDRO	474#	476#	478#	480#	482#	531#	615#	631#	1167#	1189#	1211#	1233#	1258#	1280#	1843#
	1853#	1871#	1891#	1942#	1951#	2171#	2210#	3220#							
M#MCHI	17#														
M#MCLO	17#														
M#POP	74#	99#	120#	151#	469#	601#	616#	632#	644#	656#	1144#	1169#	1191#	1213#	1235#
	1264#	1313#	1341#	1342#	1368#	1369#	1391#	1392#	1418#	1419#	1461#	1462#	1498#	1499#	1531#
	1532#	1569#	1570#	1609#	1650#	1691#	1739#	1775#	1829#	1854#	1890#	1892#	1929#	1952#	1988#
	2020#	2035#	2045#	2118#	2121#	2146#	2159#	2209#	2213#	2259#	2372#	2384#	2451#	2488#	2519#
	2566#	2693#	2695#	2769#	2771#	3222#									
M#PRIN	640#	652#	1135#	1136#	1137#	2103#	2112#	2199#	2208#	2876#	3043#	3082#	3132#	3135#	3138#
	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#	3188#	3189#	3192#	3204#	3205#
	3206#	3207#	3219#												
M#PUSH	63#	89#	105#	143#	465#	473#	605#	620#	636#	648#	1130#	1157#	1179#	1201#	1223#
	1243#	1279#	1323#	1325#	1352#	1354#	1378#	1380#	1402#	1406#	1429#	1431#	1472#	1474#	1508#
	1510#	1542#	1546#	1583#	1620#	1661#	1702#	1749#	1786#	1841#	1866#	1870#	1902#	1939#	1967#
	2005#	2009#	2024#	2039#	2050#	2133#	2137#	2150#	2164#	2227#	2278#	2296#	2398#	2459#	2495#
	2526#	2577#	2632#	2702#	2755#	3127#									
M#PUT	640#	652#	1135#	1136#	1137#	2103#	2112#	2199#	2208#	2876#	3043#	3082#	3132#	3135#	3138#
	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#	3188#	3189#	3192#	3204#	3205#
	3206#	3207#	3219#												
M#PUT1	640#	652#	1135#	1136#	1137#	2103#	2112#	2199#	2208#	2876#	3043#	3082#	3132#	3135#	3138#
	3141#	3144#	3147#	3150#	3153#	3157#	3164#	3173#	3180#	3181#	3188#	3189#	3192#	3204#	3205#
	3206#	3207#	3219#												
M#RADI	64#	66#	68#	69#	70#	71#	73#	106#	108#	110#	111#	113#	114#	115#	116#
	117#	118#	119#												
M#RNRO	531#	3435#													
M#SETS	63#	89#	105#	143#	465#	473#	605#	620#	636#	648#	1130#	1157#	1179#	1201#	1223#
	1243#	1279#	1323#	1325#	1352#	1354#	1378#	1380#	1402#	1406#	1429#	1431#	1472#	1474#	1508#
	1510#	1542#	1546#	1583#	1620#	1661#	1702#	1749#	1786#	1841#	1866#	1870#	1902#	1939#	1967#
	2005#	2009#	2024#	2039#	2050#	2133#	2137#	2150#	2164#	2227#	2278#	2296#	2398#	2459#	2495#
	2526#	2577#	2632#	2702#	2755#	3127#									
M#SVC	474#	476#	478#	480#	482#	518#	522#	531#	601#	614	615#	616#	631#	632#	640#
	644#	652#	656#	929	1135#	1136#	1137#	1138#	1144#	1166	1167#	1168#	1169#	1188	1189#
	1190#	1191#	1210	1211#	1212#	1213#	1232	1233#	1234#	1235#	1257	1258#	1259#	1263	1264#
	1280#	1300	1304	1307	1312	1313#	1325#	1336	1337#	1341#	1342#	1354#	1363	1364#	1368#
	1369#	1380#	1386	1387#	1391#	1392#	1406#	1413	1414#	1418#	1419#	1431#	1443	1444#	1456
	1457#	1461#	1462#	1474#	1484	1485#	1493	1494#	1498#	1499#	1510#	1517	1518#	1526	1527#
	1531#	1532#	1546#	1554	1555#	1564	1565#	1569#	1570#	1594	1595#	1600	1601#	1608	1609#
	1635	1636#	1641	1642#	1649	1650#	1676	1677#	1682	1683#	1690	1691#	1718	1719#	1724
	1725#	1730	1731#	1738	1739#	1765	1766#	1770	1771#	1774	1775#	1808	1809#	1816	1817#
	1818#	1827	1828#	1829#	1843#	1852	1853#	1854#	1870#	1871#	1885	1886#	1890#	1891#	1892#
	1921	1922#	1929#	1942#	1949	1950#	1951#	1952#	1980	1987	1988#	2009#	2020#	2024#	2035#

CVRLBB0 -- RLV12 DISKLESS.
CVRLBB.P11 12-JUN-85 15:33

MACY11 30(1046) 12-JUN-85 15:42 PAGE 49-5
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0158

POINTE	49															
PRINTB	2112	2208	3132	3135	3138	3141	3144	3147	3150	3153	3157	3164	3173	3204	3205	
	3206	3207														
PRINTF	640	652	1135	1136	1137	2876	3043	3082	3219							
PRINTX	2103	2199	3180	3181	3188	3189	3192									
READEF	474	476	478	480	482											
SETPRI	631	1280	1843	1853	1871	1891	1942	1951	2171	2210						
SFERR	207#	614	1166	1188	1210	1232	1257									
STARS	1151	1156	1173	1178	1195	1200	1217	1222	1239	1242	1268	1278	1317	1322	1346	
	1351	1373	1377	1396	1401	1423	1428	1466	1471	1503	1507	1536	1541	1574	1582	
	1613	1619	1654	1660	1695	1701	1743	1748	1779	1785	1833	1840	1858	1865	1896	
	1901	1934	1938	1956	1966	1992	2004	2125	2132	2217	2226	2263	2276	2389	2397	
	2455	2458	2492	2494	2523	2525	2570	2576	2699	2701						
SVC	16#	17														
XFER	67	72	1138#	1818#	2046#	2060#	2160#	2232#	2284#	2626#	2751#					
XFERF	65	109														
XFERT	112															

. ABS. 040434 000

ERRORS DETECTED: 0

CVRLBB.BIC, CVRLBB.LST/LI:TOC/CRF=SVC33/ML, CVRLBB.P11
 RUN-TIME: 33 34 4 SECONDS
 RUN-TIME RATIO: 128/72=1.7
 CORE USED: 19K (38 PAGES)