

MB061,RLV12

RLV12 DISKLESS
CVRLBA0

AH-S827A MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 1982
MADE IN USA



Microfiche grid containing multiple frames of data, likely a program listing or system output. The content is too small to transcribe accurately but appears to be a structured list or table.



IDENTIFICATION

PRODUCT CODE: AC-S825A-MC
PRODUCT NAME: CVRLBA0 RLV12 DISKLESS
DATE CREATED: FEB 1982
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: G. PASQUANTONIO

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 DIGITAL EQUIPMENT CORPORATION

1.0 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).

1.3 RELATED DOCUMENTS AND STANDARDS

 CHQUSB 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(CEED)	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.

2.2 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/TEST: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...

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.

3.0 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
```


- 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 (JUNTING)
5 = SEEK - LINEAR MODE
6 = UNLOAD HEADS
7 = SPIN DOWN

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

I 1

SEQ 0008

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.

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 IN 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.

- 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 AND /23 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.

7.13 TEST 27 -- MAINTENANCE, FIFO ADDRESS TEST (RLV12/11 ONLY). L 1

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.

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

M 1

SEQ 0012

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, USEING 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 USING 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 PSEUDO-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
51	HARDWARE PARAMETER CODING
93	SOFTWARE PARAMETER CODING
145	GLOBAL EQUATES
221	GLOBAL DATA
452	INITIALIZATION CODE
630	GLOBAL SUBROUTINES
986	MEMORY SIZER
1094	REPORT ENVIRONMENT
1115	
1116	RL DISKLESS CONTROLLER TESTS.
1117	
1118	1 -- RLCS ADDRESSABILITY.
1140	2 -- RLBA ADDRESSABILITY.
1162	3 -- RLDA ADDRESSABILITY.
1184	4 -- RLMP ADDRESSABILITY.
1206	5 -- RLBAE ADDRESSABILITY (RLV12 ONLY).
1235	6 -- BUS RESET OF ALL REGISTERS.
1284	7 -- READ WRITE OF RLCS.
1313	8 -- READ WRITE OF RLBA.
1340	9 -- READ WRITE OF RLDA.
1363	10 -- READ WRITE OF RLBAE (RLV12 ONLY).
1390	11 -- BIS AND BIC OF RLCS.
1433	12 -- BIS AND BIC OF RLBA.
1470	13 -- BIS AND BIC OF RLDA.
1503	14 -- BIS AND BIC OF RLBAE (RLV12 ONLY).
1541	15 -- UNIQUENESS OF RLCS.
1580	16 -- UNIQUENESS OF RLBA.
1621	17 -- UNIQUENESS OF RLDA.
1662	18 -- UNIQUENESS OF RLMP.
1710	19 -- UNIQUENESS OF RLBAE (RLV12 ONLY).
1746	20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).
1800	21 -- TEST INTERRUPT ON FUNCTION (0) COMPLETE.
1825	22 -- TEST INTERRUPT PRIORITY BR LEVEL.
1863	23 -- RLV11/12 MAINTENANCE, FORCED OPI (WC <> 511.)
1901	24 -- RLV11/12 MAINTENANCE, FORCED OPI INTERRUPT.
1923	25 -- RLV11/12 MAINTENANCE, OPI TIMING TEST.
1959	26 -- RLV11/12 MAINTENANCE, FIFO DMA AND CRC CHECK.
2092	27 -- RLV11/12 MAINTENANCE, FIFO ADDRESSING.
2184	28 -- RLV11/12 MAINTENANCE, BANK 7 SELECT AND NEXM TEST.
2230	29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.
2355	
2356	RLV12 DRIVE INTERFACE TESTS (G5388 TLM REQUIRED).
2357	
2361	30 -- SYS CLK, PWR OK, DRIVE SELECT, READY, AND ERROR BITS.
2422	31 -- DRIVE COMMAND, STATUS AND STATUS CLOCK.
2459	32 -- DRIVE COMMAND, SEEK DIFF AND SECTOR PULSE.
2490	33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.
2537	34 -- READ DATA, READ HEADER, AND READ DATA W/O HEADER.
2666	35 -- WRITE CHECK.
2864	TLM PROM DUMP UTILITY.

CVRLBAO -- RLV12 DISKLESS. MACY11 30G(1063) 08-FEB-82 10:07 C 2
CVRLBA.P11 08-FEB-82 09:59 TABLE OF CONTENTS

SEQ 0015

3071	
3072	SUPERVISOR DISPATCH TABLE.
3092	GLOBAL ERROR HANDLERS AND ASCII TEXT
3343	DEVELOPMENT/DEBUG AIDS
3344	
3345	RLV12 EMULATOR

14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
(4)
(4)
(4)
(4)
(4)
(4)
(6)
(6)
(5)
(5)
(4)
(5)
(4)
(5)
(4)
(5)
(4)
(5)
(4)
(5)
(4)
(5)
(4)
(5)
(4)

000000'
000001
000001
000001
000001
000001
000001
002000
000100
000000
000000
002000
002000
002000
002001 103
002002 126
002003 122
002004 114
002005 102
002006 000
002007 000
002010 000
002010 101
002011 060
002012 000001
002014 000036
002016 002172
002020 002412
002022 002376
002024 003304
002026 040344

```
.SBTTL PROGRAM HEADER  
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
      .MCALL SVC  
      SVC ; INITIALIZE SUPERVISOR MACROS  
  
SVCGBL= 1 ; LIST GLOBAL TAGS AT RIGHT MARGIN.  
SVCTST= 1 ; DITTO TEST TAGS.  
SVCSUB= 1 ; DITTO SUBTEST TAGS.  
SVCTAG= 1 ; DITTO ANY OTHER TAGS.  
SVCINS= 1 ; DITTO INSTRUCTIONS AND DATA.  
  
: THESE SYMBOLS CONTROL THE LISTING FIELD OF ALL SVC MACRO  
: EXPANSIONS. YOU MAY CHANGE THEM AT ANY TIME OR PLACE.  
  
: 1 = RIGHT-JUSTIFY (MAKES IT EASY TO DISTINGUISH  
: SVC'S MACRO CODE FROM YOUR OWN).  
: 0 = LEFT-JUSTIFY (ALIGN IN A NORMAL FASHION).  
: -1 = DON'T LIST THE EXPANSIONS AT ALL.  
:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
  
      .ENABL ABS,AMA  
      = 2000  
  
PRGSIZ= ^H<LSLAST> ; PROGRAM SIZE IN 1/8 K UNITS (OCTAL).  
SVCGBL= 0 ; ALIGN THE HEADER STUFF.  
SVCINS= 0  
  
      POINTER BGNSFT,BGNSW,BGNDU,BGNAU,BGNRPT,BGNSETUP  
      HEADER CVRLB,A,0,30,0,340  
LSNAME:: ;DIAGNOSTIC NAME  
      .ASCII /C/  
      .ASCII /V/  
      .ASCII /R/  
      .ASCII /L/  
      .ASCII /B/  
      .BYTE 0  
      .BYTE 0  
      .BYTE 0  
  
LSREV:: ;REVISION LEVEL  
      .ASCII /A/  
LSDEPO:: ;0  
      .ASCII /0/  
LSUNIT:: ;NUMBER OF UNITS  
      .WORD TSPTHV  
LSTIML:: ;LONGEST TEST TIME  
      .WORD 30.  
LSHPCP:: ;POINTER TO H.W. QUES.  
      .WORD LSHARD  
LSSPCP:: ;POINTER TO S.W. QUES.  
      .WORD LSSOFT  
LSHPTP:: ;PTR. TO DEF. H.W. PTABLE  
      .WORD LSHW  
LSSPTP:: ;PTR. TO S.W. PTABLE  
      .WORD LSSW  
LSLADP:: ;DIAG. END ADDRESS  
      .WORD LSLAST
```

(5)	002030		LSSTA::		;RESERVED FOR APT STATS
(4)	002030	000000			
(5)	002032		LSCO::	.WORD 0	
(4)	002032	000000			
(5)	002034		LSDTYP::		;DIAGNOSTIC TYPE
(4)	002034	000000			
(5)	002036		LSAPT::		;APT EXPANSION
(4)	002036	000000			
(5)	002040		LSDTP::		;PTR. TO DISPATCH TABLE
(4)	002040	026052			
(5)	002042		LSPRIO::		;DIAGNOSTIC RUN PRIORITY
(4)	002042	000340			
(5)	002044		LSENV1::		;FLAGS DESCRIBE HOW IT WAS SETUP
(4)	002044	000000			
(5)	002046		LSEXP1::		;EXPANSION WORD
(4)	002046	000000			
(5)	002050		LSMREV::		;SVC REV AND EDIT #
(4)	002050	003			
(3)	002051	003			
(5)	002052		LSEF::		;DIAG. EVENT FLAGS
(4)	002052	000000			
(5)	002054	000000			
(5)	002056		LSSPC::		
(4)	002056	000000			
(5)	002060		LSDEVP::		; POINTER TO DEVICE TYPE LIST
(4)	002060	002142			
(5)	002062		LSREPP::		;PTR. TO REPORT CODE
(4)	002062	011644			
(5)	002064		LSEXP4::		
(4)	002064	000000			
(5)	002066		LSEXP5::		
(4)	002066	000000			
(5)	002070		LSAUT::		;PTR. TO ADD UNIT CODE
(4)	002070	007432			
(5)	002072		LSDUT::		;PTR. TO DROP UNIT CODE
(4)	002072	007350			
(5)	002074		LSLUN::		;LUN FOR EXERCISERS TO FILL
(4)	002074	000000			
(5)	002076		LSDESP::		;PTR. TO DIAG. DESCRIPTION
(4)	002076	002122			
(5)	002100		LSLOAD::		;GENERATE SPECIAL AUTOLOAD EMT
(4)	002100	104035			
(5)	002102		LSETP::		;PTR. TO ERR_TBL
(4)	002102	000000			
(5)	002104		LSICP::		;PTR. TO INIT CODE
(4)	002104	006374			
(5)	002106		LSCCP::		;PTR. TO CLEAN-UP CODE
(4)	002106	007272			
(5)	002110		LSACP::		;PTR. TO AUTO CODE
(4)	002110	007214			
(5)	002112		LSPRT::		;PTR. TO PROTECT TABLE
(4)	002112	006366			
(5)	002114		LSTEST::		;TEST NUMBER
(4)	002114	000000			
(5)	002116		LSDLY::		;DELAY COUNT
(4)	002116	000000			

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 F 2
PROGRAM HEADER PAGE 2-2

SEG 0018

(5)	002120				LSHIME::	.WORD 0	:PTR. TO HIGH MEM
(4)	002120	000000					
43							
44	002122				LSDESC::	DESCRIPT <RLV12 DISKLESS>	
(4)	002122						
(3)	002122	046122	030526	020062		.ASCIZ /RLV12 DISKLESS/	
(3)	002130	044504	045523	042514			
(3)	002136	051523	000				
(2)		002142				.EVEN	
45							
46	002142				LSDVTYP::	DEVTYP <RLV12, RLV11, OR RL11>	
(4)	002142						
(3)	002142	046122	030526	026062		.ASCIZ /RLV12, RLV11, OR RL11/	
(3)	002150	051040	053114	030461			
(3)	002156	020054	051117	051040			
(3)	002164	030514	000061				
(2)						.EVEN	
47							
48		000001			SVCGBL= 1		: SHOVE EVERYTHING BACK TO THE RIGHT.
49		000001			SVCINS= 1		

```
51          .SBTTL  HARDWARE PARAMETER CODING
52          .:
53          .: GET PARAMETERS FROM OPERATOR.
54          .:
55          BGNHRD
56          GPRML  RL1,10,BIT1,YES          ; RLV12 ??          LSHARD:: .WORD L10000-LSHARD/2
57          XFERF  1$
58          GPRML  RL1A,10,BIT0,YES        ; BAE ENABLED ??    .WORD TSCODE
59          XFER   2$
60          GPRML  RL2,10,BIT0,YES        ; RLV11 ??          .WORD TSCODE
61          GPRMA  CSRA,0,0,160000,177776,YES ; CSR.              .WORD TSCODE
62          GPRMA  VECA,2,0,0,776,YES     ; VECTOR.          .WORD CSRA
63          GPRMD  BRL,4,0,340,0,7,YES    ; BR LEVEL.       .WORD T$LOLIM
64          XFER   3$                      ; DRIVE IS ALWAYS 0 IN DISKLESS. .WORD TSHILIM
65          GPRMD  DRN,6,0,3400,0,7,YES   ; DRIVE.          .WORD TSCODE
66          ENDHRD
67          L10000: .EVEN
68          .: THESE ARE THE HARDWARE QUERIES...
69          RL1:   .ASCIZ 'RLV12          :
70          RL1A: .ASCIZ 'BAE ENABLED   :
71          RL2:   .ASCIZ 'RLV11          :
72          046122 030526 004462
73          040502 020105 047105
74          122   053114 030461
```


CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 H 2
HARDWARE PARAMETER CODING PAGE 3-1

SEQ 0020

73	002323	103	051123	040440	CSRA:	.ASCIZ	'CSR ADDRESS	:
74	002340	042526	052103	051117	VECA:	.ASCIZ	'VECTOR	:
75	002351	102	020122	042514	BRL:	.ASCIZ	'BR LEVEL	:
76	002363	104	044522	042526	DRN:	.ASCIZ	'DRIVE	:
77		002374				.EVEN		: UNUSED.

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

81	002374	
(3)	002374	000005
(3)	002376	
82	002376	174400
83	002400	000160
84	002402	000200
85	002404	000000
86	002406	000003
87		
88		
89		
90		
91	002410	
(3)	002410	

BGNHW

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

LSHW:: .WORD L10001-LSHW/2

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

ENDHW

L10001:

```
93          .SBTTL SOFTWARE PARAMETER CODING
94          :
95          : GET SOFT PARAMETERS FROM OPERATOR.
96          :
97          BGNSFT
98          (3) 002410 000054          .WORD L10002-LSSOFT/2
99          (3) 002412          .WORD LSSOFT::
100         (4) 002412 000052          GPRMD ADLIM,0,D,-1,0,-1,YES ; GET AUTO-DROP ERROR LIMIT.
101         (4) 002414 002545          .WORD TSCODE
102         (4) 002416 177777          .WORD ADLIM
103         (4) 002420 000000          .WORD -1
104         (4) 002422 177777          .WORD TSLOLIM
105         99 002424          .WORD TSHILIM
106         (4) 002424 000003          DISPLAY TLMBP ; BYPASS MESSAGE.
107         (4) 002426 002575          .WORD TSCODE
108         100 002430          .WORD TLMBP
109         (4) 002430 005130          GPRML TLMIN,12,BIT0,YES ; TLM ON THIS UNIT ??
110         (4) 002432 002751          .WORD TSCODE
111         (4) 002434 000001          .WORD TLMIN
112         101 002436          .WORD BIT0
113         (5) 002436 042044          XFERF 1$
114         102 002440          .WORD TSCODE
115         (4) 002440 006031          GPRMA TCSU0,14,0,160000,177776,YES ; TLM CSR FOR UNIT 0.
116         (4) 002442 002776          .WORD TSCODE
117         (4) 002444 160000          .WORD TCSU0
118         (4) 002446 177776          .WORD TSLOLIM
119         103 002450          .WORD TSHILIM
120         (4) 002450 001130          GPRML PRMDMP,2,BIT0,YES ; PROM DUMP ??
121         (4) 002452 003024          .WORD TSCODE
122         (4) 002454 000001          .WORD PRMDMP
123         104 002456          .WORD BIT0
124         (5) 002456 032024          XFERT 1$
125         105 002460          .WORD TSCODE
126         (4) 002460 002032          GPRMD PRM,4,0,-1,0,-1,YES ; PROM ID.
127         (4) 002462 003047          .WORD TSCODE
128         (4) 002464 177777          .WORD PRM
129         (4) 002466 000000          .WORD -1
130         (4) 002470 177777          .WORD TSLOLIM
131         106 002472          .WORD TSHILIM
132         (4) 002472 003130          GPRML INHMP,6,BIT15,YES ; INHIBIT MAX PEAK...
133         (4) 002474 003070          .WORD TSCODE
134         (4) 002476 100000          .WORD INHMP
135         107 002500          .WORD BIT15
136         (4) 002500 003130          GPRML INHMNP,6,BIT14,YES ;...MIN PEAK...
137         (4) 002502 003120          .WORD TSCODE
138         (4) 002504 040000          .WORD INHMNP
139         108 002506          .WORD BIT14
140         (4) 002506 003130          GPRML INHNC,6,BIT2,YES ;...NOMINAL CLOCK...
141         (4) 002510 003150          .WORD TSCODE
142         (4) 002512 000004          .WORD INHNC
143         109 002514          .WORD BIT2
144         (4) 002514 003130          GPRML INHFC,6,BIT1,YES ;...FAST CLOCK...
145         (4) 002516 003177          .WORD TSCODE
146         (4) 002520 000002          .WORD INHFC
147         110 002522          .WORD BIT1
```


(4) 002522 003130
(4) 002524 003223
(4) 002526 000001
111 002530
(4) 002530 004032
(4) 002532 003247
(4) 002534 000007
(4) 002536 000000
(4) 002540 000006
112 002542
(2)
(3) 002542

GPRMD SEC1,10,0,7,0,6,YES ; SINGLE SECTOR OPTION.

.WORD TSCODE
.WORD INHSC
.WORD BIT0
.WORD TSCODE
.WORD SEC1
.WORD 7
.WORD TSLOLIM
.WORD TSHILIM

1\$: ENDSFT

.EVEN

L10002:

;; THESE ARE THE SOFTWARE QUERIES...

113
114
115
116 002542 051105 047522 020122 ADLIM: .ASCIZ 'ERROR LIMIT FOR AUTO-DROP '
117
118 002575 10: 046114 051040 TLMBP: .ASCIZ 'ALL REMAINING QUERIES ARE FOR OPTIONAL (MANUFACTURING)'
119 002664 032507 034063 020070 .ASCIZ 'G5388 TEST-LOOP-MODULE SET-UP. USE <^Z> TO BYPASS.'
120 002750 000 .BYTE 0
121 002751 107 031465 034070 TLMIN: .ASCIZ 'G5388 TLM INSTALLED '
122 002776 051503 020122 042101 TCSUO: .ASCIZ 'CSR ADDRESS (UNIT 0) '
123 003024 052504 050115 050040 PRMDMP: .ASCIZ 'DUMP PROM ON TTY: '
124 003047 120 047522 020115 PRM: .ASCIZ 'PROM ID NUMBER '
125 003070 047111 044510 044502 INHMXP: .ASCIZ 'INHIBIT MAX PEAK SHIFT '
126 003120 047111 044510 044502 INHMNP: .ASCIZ 'INHIBIT MIN PEAK SHIFT '
127 003150 047111 044510 044502 INHNC: .ASCIZ 'INHIBIT NOMINAL CLOCK '
128 003177 111 044116 041111 INHFC: .ASCIZ 'INHIBIT FAST CLOCK '
129 003223 111 044116 041111 INHSC: .ASCIZ 'INHIBIT SLOW CLOCK '
130 003247 123 041505 047524 SEC1: .ASCIZ 'SECTOR NUMBER (0=USE ALL)'
131 003302 .EVEN

;; ... AND THESE ARE THE DEFAULT SOFT PARAMETERS.

132
133
134
135 003302
(3) 003302 000007
(3) 003304
136 003304 000000
137 003306 000000
138 003310 000401
139 003312 000000
140 003314 000000
141 003316 000000
142 003320 160010
143 003322
(3) 003322

BGNSW

ERRLMT: .WORD 0
PDSW: .WORD 0
PROMID: .WORD 401
MPXCLK: .WORD 0
SNGLSEC: .WORD 0
TLMF: .WORD 0
TCSO: .WORD 160010
ENDSW

LSSW:: .WORD L10003-LSSW/2
:: (00) NZ = DROP AFTER N ERRORS.
:: (02) PROM DUMP SWITCH (BIT0).
:: (04) CURRENT PROM SET ID.
:: (06) PEAK SHIFT AND CLOCK INHIBITS.
:: (10) SINGLE SECTOR TO USE (0 = ALL 6).
:: (12) G5388 TLM FLAG (1 = INSTALLED).
:: (14) G5388 TLM CSR ADDRESS.

L10003:


```

(1)          000004      ÉVL==          4
(1)          000010      LOT==          10
(1)          000020      ADR==          20
(1)          000040      IDU==          40
(1)          000100      ISR==         100
(1)          000200      UAM==         200
(1)          000400      BOE==         400
(1)          001000      PNT==        1000
(1)          002000      FRI==        2000
(1)          004000      IXE==        4000
(1)          010000      IBE==       10000
(1)          020000      IER==       20000
(1)          040000      LOE==       40000
(1)          100000      HOE==      100000
148          :
149          : RL/RLV EQUATES.
150          :
151          000004      ÉRRVEC= 4      : BUS-ERROR VECTOR.
152          000000      RL11= 0      : EQUATE THE CONTROLLER TYPES.
153          000001      RLV11= 1
154          000002      RLV12= 2      : RLV12 WITH BAE DISABLED.
155          000003      RLV12X= 3     : RLV12 WITH BAE ENABLED.
156
157          100000      ERR= BIT15     : COMPOSITE ERROR (RLCS)
158          040000      DERR= BIT14     : DRIVE ERROR (RLCS)
159          020000      NXM= BIT13     : NON-EXISTANT MEMORY (IF OPI = 0)...
160          020000      PAR= BIT13     : ...OR PARITY ERROR (IF OPI = 1).
161          010000      DLT= BIT12     : DATA LATE ERROR (IF OPI=0)...
162          010000      HNF= BIT12     : ...OR HEADER NOT FOUND (IF OPI=1).
163          004000      DCRC= BIT11     : DATA CRC ERROR (IF OPI=0)...
164          004000      HCRC= BIT11     : ...OR HEADER CRC ERROR (IF OPI=1).
165          002000      OPI= BIT10     : OPERATION INCOMPLETE (RLCS)
166          000000      DS0= 0        : DRIVE SELECT 0 (RLCS)
167          000400      DS1= BIT8      : DRIVE SELECT 1 (RLCS)
168          001000      DS2= BIT9      : DRIVE SELECT 2 (RLCS)
169          001400      DS3= BIT8!BIT9 : DRIVE SELECT 3 (RLCS)
170          000200      CRDY= BIT7      : CONTROLLER READY (RLCS)
171          000100      INTEN= BIT6     : INTERRUPT ENABLE (RLCS)
172          000000      NOOP= 0        : (0) NO-OP -- RL11 ONLY.
173          000000      MAINT= 0       : (0) MAINTENANCE -- RLV11 AND RLV12.
174          000002      WRCHK= BIT1     : (1) WRITE CHECK FUNCTION
175          000004      GSTAT= BIT2     : (2) GET STATUS FUNCTION
176          000006      SEEK= BIT2!BIT1 : (3) SEEK FUNCTION
177          000010      RDHDR= BIT3     : (4) READ HEADER FUNCTION
178          000012      WRITE= BIT3!BIT1 : (5) WRITE DATA FUNCTION
179          000014      READ= BIT3!BIT2  : (6) READ DATA FUNCTION
180          000016      RDNHDR= BIT3!BIT2!BIT1 : (7) READ DATA WITHOUT HEADER CHECK.
181          000001      DRDY= BIT0     : DRIVE READY.
182
183          000010      DRST= BIT3      : DRIVE RESET (RLDA)
184          000002      GSBIT= BIT1     : GET STATUS BIT (RLDA)
185          000001      MK= BIT0       : MARKER BIT (RLDA)
186          000004      SIGN= BIT2     : SIGN BIT (RLDA)
187          000100      RHHS= BIT6     : HEAD SELECT IN READ HEADER
188          000100      STHS= BIT6     : HEAD SELECT IN STATUS BACK

```

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 ^{M 2} PAGE 5-2
GLOBAL EQUATES

SEQ 0025

189

000020

DAHS= BIT4

;HEAD SELECT IN SEEK

191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219

000000
000143

000401
000402
000403

```

:
: A COUPLE OF MACROS TO REDEFINE THE ERROR CALLING CONVENTIONS
: SO THAT SEQUENTIAL ERRORS ARE ASSIGNED AT ASSEMBLY TIME,
: AND I DON'T HAVE TO SCREW AROUND WITH THEM !!
:
SFN= 0 ; SYSTEM FATAL ERRORS RUN FROM 1 TO 99.
DFN= 99. ; DEVICE FATAL ERRORS RUN FROM 100. UP.
:
.MACRO SFERR ADDR,PNTR
SFN=SFN+1
ERRSF SFN,ADDR,PNTR
.ENDM
:
.MACRO DFERR ADDR,PNTR
DFN=DFN+1
ERRDF DFN,ADDR,PNTR
.ENDM
:
: ANOTHER TO START EACH TEST.
:
.MACRO BEGIN.TEST
BGNTST
.ENDM
:
: AND A COUPLE OF HANDY SKIPS.
:
SKP1= BR+1
SKP2= BR+2
SKP3= BR+3

```

221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276

.SBTTL GLOBAL DATA

UNITST: .WORD 0 ; CURRENT UNIT UNDER TEST.
RLCS: .WORD 0 ; REGISTER ADDRESSES.
RLBA: .WORD 0
RLDA: .WORD 0
RLMP: .WORD 0
RLBAE: .WORD 0
BCSR: .WORD 0 ; BASE ADDRESS
BPRIOR: .WORD 0
BVEC: .WORD 0 ; G5388 TLM CSR ADDRESS.
TCSR: .WORD 0
DRIVE: .WORD 0 ; REGISTERS BEFORE FUNCTION.
B.CS: .WORD 0
B.BA: .WORD 0
B.DA: .WORD 0
B.MP: .WORD 0
B.BAE: .WORD 0
DERFLG: .WORD 0
E.CS: .WORD 0 ; REGISTERS AFTER FUNCTION.
E.BA: .WORD 0
E.DA: .WORD 0
E.MP: .WORD 0
E.MP1: .WORD 0 ; MAINT MODE CRC CHECK ONLY.
E.BAE: .WORD 0
CPUTYP: .WORD 0 ; 0 = 11/03, +NZ = 11/23, -1 = UNKNOWN.P
RLTYP: .WORD 0 ; RL TYPE -- 0 = RL11, 1 = RLV11, 2(3) = RLV12.
XPOLY: .WORD 120001
BCCFBK: .WORD 0 ; LOCATION USED BY "SIMBCC"
CALBCC: .WORD 0 ; LOCATION USED BY "SIMBCC"
TEMP2: .WORD 0 ; LOCATION USED BY "SIMBCC"
TEMP3: .WORD 0 ; LOCATION USED BY "SIMBCC"
TEMP4: .WORD 0 ; LOCATION USED BY "SIMBCC"
TEMP5: .WORD 0 ;
TMP0: .WORD 0 ; DITTO
TMP1: .WORD 0 ; DITTO
TMP2: .WORD 0
GDDAT: .WORD 0
BDDAT: .WORD 0
INIMP: .WORD 0 ; HOLDS INITIAL MP WORD COUNT.
INIDA: .WORD 0 ; HOLDS CURRENT DA TEST WORD.
GDCRC3: .WORD 0 ; HOLDS CRC OF DA+3
GDCRC4: .WORD 0 ; HOLDA CRC OF CRC OF DA+4
OPIMN: .WORD 155.
OPIMX: .WORD 650.
HINUM: .WORD 176543
LONUM: .WORD 123456
TENLO: .WORD 0
TENHI: .WORD 0
DLYCNT: .WORD 0 ; 1 MSEC DELAY COUNTER.
DROPPED: .WORD 0 ; UNITS DROPPED COUNTER.
ERPOINT: .WORD 0 ; POINTS TO ONE OF THE FOLLOWING:
ERCOUNT: .BLKW 64. ; ERROR COUNTERS (ENOUGH FOR 64 UNITS).
; PATTERNS USED FOR LOADING/READING REGISTERS

277				
278	003670	000000	BEGPAT: 000000	:GROWING 1
279	003672	000001	000001	
280	003674	000003	000003	
281	003676	000007	000007	
282	003700	000017	000017	
283	003702	000037	000037	
284	003704	000077	000077	
285	003706	000177	000177	
286	003710	000377	000377	
287	003712	000777	000777	
288	003714	001777	001777	
289	003716	003777	003777	
290	003720	007777	007777	
291	003722	017777	017777	
292	003724	037777	037777	
293	003726	077777	077777	
294	003730	177777	177777	
295	003732	177776	177776	:GROWING 0
296	003734	177774	177774	
297	003736	177770	177770	
298	003740	177760	177760	
299	003742	177740	177740	
300	003744	177700	177700	
301	003746	177600	177600	
302	003750	177400	177400	
303	003752	177000	177000	
304	003754	176000	176000	
305	003756	174000	174000	
306	003760	170000	170000	
307	003762	160000	160000	
308	003764	140000	140000	
309	003766	100000	100000	
310	003770	000000	000000	
311	003772	000001	000001	:WALKING 1
312	003774	000002	000002	
313	003776	000004	000004	
314	004000	000010	000010	
315	004002	000020	000020	
316	004004	000040	000040	
317	004006	000100	000100	
318	004010	000200	000200	
319	004012	000400	000400	
320	004014	001000	001000	
321	004016	002000	002000	
322	004020	004000	004000	
323	004022	010000	010000	
324	004024	020000	020000	
325	004026	040000	040000	
326	004030	100000	100000	
327	004032	177777	177777	:WALKING 0
328	004034	177776	177776	
329	004036	177775	177775	
330	004040	177773	177773	
331	004042	177767	177767	
332	004044	177757	177757	

333 004046 177737
334 004050 177677
335 004052 177577
336 004054 177377
337 004056 176777
338 004060 175777
339 004062 173777
340 004064 167777
341 004066 157777
342 004070 137777
343 004072 077777
344 004074 177777
345 004076 000000

177737
177677
177577
177377
176777
175777
173777
167777
157777
137777
077777
177777

ENDPAT: 000000

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

349 004100 000000
350 004102 000002
351 004104 000004
352 004106 000010
353 004110 000020
354 004112 000040
355 004114 000100
356 004116 000400
357 004120 001000
358 004122 001576
359 004124 001574
360 004126 001570
361 004130 001560
362 004132 001540
363 004134 001500
364 004136 001400
365 004140 001576
366 004142 001574
367 004144 001566
368 004146 001556
369 004150 001536
370 004152 001436
371 004154 001136
372 004156 000076
373 004160 000006
374 004162 000016
375 004164 000036
376 004166 000076
377 004170 000176
378 004172 000576
379 004174 001576
380 004176 000000

CSPAT: .WORD 0 ;SHIFTING 1
.WORD BIT1
.WORD BIT2
.WORD BIT3
.WORD BIT4
.WORD BIT5
.WORD BIT6
.WORD BIT8
.WORD BIT9
.WORD 1576 ;GROWING 0
.WORD 1574
.WORD 1570
.WORD 1560
.WORD 1540
.WORD 1500
.WORD 1400
.WORD 1576 ;SHIFT 0
.WORD 1574
.WORD 1566
.WORD 1556
.WORD 1436
.WORD 1136
.WORD 76
.WORD 6 ;GROWING 1
.WORD 16
.WORD 36
.WORD 76
.WORD 176
.WORD 576
.WORD 1576
CSEND: .WORD 0

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

384 004200 155552
385 004202 155530
386 004204 066663
387 004206 125247
388 004210 052522

PATCRC: 155552
155530
066663
125247
052522

389	004212	177774	177774
390	004214	000374	000374
391	004216	022217	022217
392	004220	044441	044441
393	004222	166663	166663
394	004224	144441	144441
395	004226	033330	033330
396	004230	0111106	0111106
397	004232	070704	070704
398	004234	107065	107065
399	004236	1111106	1111106
400	004240	167353	167353
401	004242	156732	156732
402	004244	146311	146311
403	004246	135670	135670
404	004250	114626	114626
405	004252	104205	104205
406	004254	073564	073564
407	004256	063143	063143
408	004260	042101	042101
409	004262	031460	031460
410	004264	021037	021037
411	004266	010416	010416
412	004270	000000	000000
413			
414			
415			
416	004272	155555	155555
417	004274	133333	133333
418	004276	066666	066666
419	004300	125252	125252
420	004302	052525	052525
421	004304	177777	177777
422	004306	000000	000000
423	004310	107070	107070
424	004312	070707	070707
425	004314	144444	144444
426	004316	033333	033333
427	004320	011111	011111
428	004322	022222	022222
429	004324	044444	044444
430	004326	111111	111111
431	004330	166666	166666
432	004332	010421	010421
433	004334	021042	021042
434	004336	031463	031463
435	004340	042104	042104
436	004342	063146	063146
437	004344	073567	073567
438	004346	104210	104210
439	004350	114631	114631
440	004352	135673	135673
441	004354	146314	146314
442	004356	156735	156735
443	004360	167356	167356
444	004362	000000	000000

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

PATDAT: 155555
...
ENDDAT: 000000 ; 28 OF THESE TOO.

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 F 3
PAGE 6-4
GLOBAL DATA

SEQ 0031

445
446
447
448 004364 000400
449 005364 000400
450 006364 000000

:
: BUFFERS FOR RLV11 MAINTENANCE FUNCTION
:
: BUF1: .BLKW 256. : MAINTENANCE MODE, DMA BUF1 => FIFO...
: BUF2: .BLKW 256. : ...AND FIFO => BUF2.
: BUFEND: 0


```
452 .SBTTL  INITIALIZATION CODE
453
454 .: LOAD DEVICE PROTECTION NOT REQUIRED.
455 .:
456 006366      BGNPROT
457 (3) 006366      LSPROT::
458 006366 177777 .WORD -1      ;CSR OFFSET MAKE NOP
459 006370 177777 .WORD -1      ;MASS BUS OFFSET MAKE NOP
460 006372 177777 .WORD -1      ;DRIVE OFFSET MAKE NOP
461 006374      ENDPROT
462 .:
463 .: ENTER HERE FROM DRS ON START, RESTART, CONTINUE, ETC...
464 .:
465 006374      BGNINIT
466 (3) 006374      L$INIT::
467 006374 012700 000034 READEF #EF.PWR      ; POWER UP ??
468 (3) 006400 104447      MOV TRAP #EF.PWR,RO
469 006402      BCOMPLETE RESTART      CSREFG
470 (2) 006402 103423      BCS RESTART
471 006404      READEF #EF.START      ; START ??
472 (3) 006404 012700 000040      MOV TRAP #EF.START,RO
473 (3) 006410 104447      BCOMPLETE START      CSREFG
474 (2) 006412 103417      BCS START
475 006414      READEF #EF.RESTART      ; RESTART ??
476 (3) 006414 012700 000037      MOV TRAP #EF.RESTART,RO
477 (3) 006420 104447      BCOMPLETE RESTART      CSREFG
478 (2) 006422 103413      BCS RESTART
479 006424      READEF #EF.NEW      ; NEW PASS ??
480 (3) 006424 012700 000035      MOV TRAP #EF.NEW,RO
481 (3) 006430 104447      BCOMPLETE NEWPAS      CSREFG
482 (2) 006432 103472      BCS NEWPAS
483 006434      READEF #EF.CONTINUE      ; CONTINUE ??
484 (3) 006434 012700 000036      MOV TRAP #EF.CONTINUE,RO
485 (3) 006440 104447      BCOMPLETE 1$      CSREFG
486 (2) 006442 103401      BCS 1$
487 006444 000506      BR NXTU      ; NONE OF THE ABOVE.
488 006446 000137 007050 1$: JMP CONT
489 .:
490 006452      START:RESTART:
491 006452 000005 RESET
492 006454 005737 003306 TST PDSW      ; PROM DUMP ??
493 006460 001402 BEQ 1$      ; BR IF NOT.
494 006462 000137 024606 JMP RONDUMP   ; *** PROM DUMPER, NO RETURN ***
495 006466 012737 176543 003452 1$: MOV #176543,HINUM ;RANDOM GEN. PRIMES.
496 006474 012737 123456 003454 MOV #123456,LONUM
497 006502 005037 003464 CLR DROPPED   ; CLEAR UNITS DROPPED COUNTER.
498 005506 012701 003470 MOV #PERCOUNT,R1 ; GET ERROR COUNTER POINTER.
499 006512 012700 000100 MOV #64,RO
500 006516 005021 2$: CLR (R1)+      ; CLEAR ALL ERROR COUNTERS.
501 006520 005300 DEC RO
502 006522 001375 BNE 2$
```

491	006524	013746	000004		ENVIRN:	MOV	@#4, -(SP)	:	SAVE VECTOR 4...
492	006530	013746	000010			MOV	@#10, -(SP)	:	...AND 10...
493	006534	010605				MOV	SP, R5	:	...AND THE STACK POINTER.
494	006536	012737	006574	000004		MOV	#LSI, @#4	:	SET BUS TRAP.
495	006544	005000				CLR	RO	:	
496	006546	005737	177776			TST	@#177776	:	PSW ADDRESS VALID ??
497	006552	000240				NOP		:	TRAP THRU 4 IF NOT (LSI11 OR 11/2).
498	006554	012737	006570	000010		MOV	#NLSI, @#10	:	SET ILLEGAL TRAP.
499		000007				MFPT=	7	:	
500	006562	000007				MFPT		:	GET PROCESSOR TYPE (IF POSSIBLE).
501	006564	000240				NOP		:	TRAP THRU 10 IF NOT (NOT 11/23).
502	006566	000402				SKP2		:	
503	006570	012700	177777		NLSI:	MOV	#-1, RO	:	CPU TYPE UNKNOWN.
504	006574	010037	003400		LSI:	MOV	RO, CPUTYP	:	CPU IS LSI SOMETHING 'R OTHER.
505	006600	010506				MOV	R5, SP	:	RESTORE STACK...
506	006602	012637	000010			MOV	(SP)+, @#10	:	...AND VECTORS.
507	006606	012637	000004			MOV	(SP)+, @#4	:	
508	006612	004737	011242			JSR	PC, .SIZE	:	SIZE AVAILABLE MEMORY.
509	006616					DORPT		:	REPORT OPERATING ENVIRONMENT.
(3)	006616	104424						TRAP	CSDRPT
510									
511	006620	023737	003464	002012	NEWPAS:	CMP	DROPPED, LSUNIT	:	UNITS STILL ALIVE ??
512	006626	002401				BLT	1\$:	
513	006630					DOCLN		:	NO, ABORT
(3)	006630	104444						TRAP	CSDCLN
514	006632	012737	003466	003466	1\$:	MOV	#ERCOUNT-2, ERPOINT	:	INIT THE UNIT ERROR POINTER.
515	006640	013737	003454	003456		MOV	LONUM, TEMLO	:	
516	006646	013737	003452	003460		MOV	HINUM, TEMHI	:	NEW PRIMES FOR NEW PASS.
517	006654	012737	177777	003322		MOV	#-1, UNITST	:	RESET UNIT NUMBER.
518	006662	005237	003322		NXTU:	INC	UNITST	:	BUMP UNIT NUMBER...
519	006666	062737	000002	003466		ADD	#2, ERPOINT	:	...AND ERROR COUNT POINTER.
520	006674	023737	003322	002012		CMP	UNITST, LSUNIT	:	
521	006702	002346				BGE	NEWPAS	:	
522	006704				1\$:	GPHARD	UNITST, RO	:	
(3)	006704	013700	003322					MOV TRAP	UNITST, RO
(3)	006710	104442							CSGPHRD
523	006712					BNCOMPLETE	NXTU	:	BR IF UNIT NOT THERE (DROPPED).
(2)	006712	103363						BCC	NXTU
524	006714	012037	003336		2\$:	MOV	(RO)+, BCSR	:	SET UP RUN TIME P-TABLE...
525	006720	005737	036704			TST	MIMIC	:	***** IF MIMIC
526	006724	001403				BEQ	3\$:	*****
527	006726	012737	037224	003336		MOV	#DUMMY, BCSR	:	***** USE DUMMY REGISTERS
528	006734	012037	003342		3\$:	MOV	(RO)+, BVEC	:	...FOR THIS UNIT.
529	006740	012037	003340			MOV	(RO)+, BPRIOR	:	
530	006744	012037	003346			MOV	(RO)+, DRIVE	:	
531	006750	012037	003402			MOV	(RO)+, RLTYP	:	SET CONTROLLER TYPE.
532									
533	006754	013737	003316	003344		MOV	TLMF, TCSR	:	RUNNING WITH TLM'S ??
534	006762	001412				BEQ	5\$:	BR IF NOT.
535	006764	013737	003320	003344		MOV	TCSO, TCSR	:	YES, GET BASE TLM CSR...
536	006772	013700	003322			MOV	UNITST, RO	:	...AND CURRENT UNIT NUMBER.
537	006776	001404				BEQ	5\$:	BR IF UNIT IS ZERO.
538	007000	062737	000010	003344	4\$:	ADD	#10, TCSR	:	OTHERWISE, ADJUST FOR CURRENT UNIT.
539	007006	077004				SOB	RO, 4\$:	
540									
541	007010	005737	003400		5\$:	TST	CPUTYP	:	


```
542 007014 003004          BGT      6$          : BR IF 11/23
543 007016 001406          BEQ      7$          : BR IF LSI 11 OR 11/2.
544 007020 012700 000764      MOV      #500.,R0    : IF NEITHER, DELAY IS UNCALIBRATED.
545 007024 000405          BR       8$          :
546 007026 012700 000443      6$: MOV      #291.,R0    : 1MS DELAY FOR 11/23.
547 007032 000402          SKP2
548 007034 012700 000170      7$: MOV      #120.,R0    : 1MS DELAY FOR LSI11 OR 11/2.
549 007040 010037 003462      8$: MOV      R0,DLYCNT : SET DELAY COUNTER.
550 007044 004737 026000      JSR      PC,ADJTN   : ADJUST THE NUMBER OF TESTS TO RUN...
551                                     : ...ACCORDING TO CURRENT CONFIGURATION.
552 007050 013737 003456 003454  CONT: MOV      TEMLO,LONUM : RESTORE RANDOM FOR NEXT UUT
553 007056 013737 003460 003452  MOV      TEMHI,HINUM : RESTORE PRIME FOR NEXT UUT
554 007064 012737 004364 010352  MOV      #BUF1,BA16 : INIT 16 BIT BUFFER ADDRESS.
555 007072 005037 010354          CLR      BA22       : ...AND 6 BIT EXTENSION (22 BITS).
556 007076 013700 003336          MOV      BCSR,R0
557 007102 010037 003324          MOV      R0,RLCS   : SET UNIT'S ADDRESSES.
558 007106 062700 000002          ADD      #2,R0
559 007112 010037 003326          MOV      R0,RLBA
560 007116 062700 000002          ADD      #2,R0
561 007122 010037 003330          MOV      R0,RLDA
562 007126 062700 000002          ADD      #2,R0
563 007132 010037 003332          MOV      R0,RLMP
564 007136 062700 000002          ADD      #2,R0
565 007142 010037 003334          MOV      R0,RLBAE
566 007146 013700 003342          MOV      BVEC,R0
567 007152 012720 010632          MOV      #RLINT,(R0)+ : SET RL VECTOR.
568 007156 013710 003340          MOV      BPRIOR,(R0)
569 007162 012737 010624 000004  MOV      #TRAP4,ERRVEC : SET TIME-OUT TRAP CATCHER.
570 007170 012737 000340 000006  MOV      #PRI07,ERRVEC+2
571 007176 012737 010636 000100  MOV      #CLKRT1,@#100 : NULL THE CLOCK VECTOR.
572 007204 012737 000340 000102  MOV      #PRI07,@#102
573 007212          ENDINIT
(3) 007212          L10005: TRAP      CSINIT
(3) 007212 104411
574                                     :
575                                     : AUTO-DROP IF FLA:ADR AND UNIT DOESN'T RESPOND.
576                                     :
577 007214          BGNAUTO
(3) 007214          LSAUTO::
578 007214 023737 003322 002012  CMP      UNITST,LSUNIT : VALID UNIT NUMBER ??
579 007222 002022          BGE      1$          : NO, DON'T TRY IT !!!
580 007224 005037 010626          CLR      TRPFLG      : CLEAR TRAP FLAG.
581 007230 005777 174070          TST      @RLCS       : TRY TO ACCESS CONTROLLER.
582 007234 000240          240
583 007236 005737 010626          TST      TRPFLG      : DID IT TRAP ??
584 007242 001412          BEQ      1$          : NO, WE'RE OK.
585 007244 013737 003324 003432  MOV      RLCS,GDDAT
586 007252          SFERR      EMO,ERR1 : REPORT NO CONTROLLER...
(5) 007252 104454          TRAP      CSERSF
(6) 007254 000001          .WORD    1
(6) 007256 030636          .WORD    EMO
(6) 007260 026170          .WORD    ERR1
587 007262          DODU      UNITST    : ...AND TELL SUPER TO DROP UNIT.
(3) 007262 013700 003322          MOV      UNITST,R0
(3) 007266 104451          TRAP      CSODDU
588 007270          1$: ENDAUTO
```



```
(3) 007270
(3) 007270 104461
589
590
591
592 007272
(3) 007272
593 007272 012700 000012
594 007276 032777 000200 174020 1$: MOV #10.,RO
595 007304 001004 BNE #CRDY,@RLCS : GIVE HIM ABOUT 100 MSEC...
596 007306 004537 010160 JSR R5,WDELAY : ...TO FINISH UP.
597 007312 000012 10.
598 007314 077010 SOB RO,1$
599 007316 012777 000200 174000 2$: MOV #CRDY,@RLCS : THEN CLEAR ALL OPTION BITS.
600 007324 023727 003402 000003 CMP RLTP,#RLV12X : BAE IN USE ??
601 007332 001002 BNE 3$
602 007334 005077 173774 CLR @RLBAE : IF SO, MAKE SURE IT'S CLEAR.
603 007340 3$: SETPRI #PRI07
(3) 007340 012700 000340
(3) 007344 104441
604 007346
(3) 007346
(3) 007346 104412
605
606
607
608 007350
(3) 007350
609 007350 023737 002012 003464
610 007356 001424
611 007360 005237 003464
612 007364
(7) 007364 012746 007406
(6) 007370 012746 000001
(3) 007374 010600
(4) 007376 104417
(4) 007400 062706 000004
613 007404 000411
614 007406 040445 047125 052111 1$: BR 2$
615 007430 .ASCIZ /%AUNIT DROPPED%/
616 007430 2$: .EVEN
(3) 007430
(3) 007430 104453
617
618
619
620 007432
(3) 007432
621 007432 005737 003464
622 007436 001423
623 007440 005337 003464
624 007444
(7) 007444 012746 007466
(6) 007450 012746 000001
(3) 007454 010600
(4) 007456 104417
```

L10006: TRAP CSAUTO

... CLEAN-UP CODING SECTION (DO AT END-PASS (OR SUB PASS)).

BGNCLN

L\$CLEAN::

ENDCLN

L10007: TRAP CSCLEAN

... DROP UNITS -- AND KEEP TRACK OF 'EM.

BGNDU

L\$DU::

L10010: TRAP C\$DU

... IF THEY'RE PICKED UP AGAIN, TWEAK THE COUNTER.

BGNAU

L\$AU::

MOV #1,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP C\$PNTF

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 ^{K 3} PAGE 7-4
INITIALIZATION CODE

SEQ 0036

(4) 007460 062706 000004
625 007464 000410
626 007466 040445 047125
627 007506
628 007506
(3) 007506
(3) 007506 104452

052111 1S: BR 2S
.ASCIZ /%AUNIT ADDED%N/
EVEN
ENDAU
2S:

ADD #4,SP

L10011: TRAP CSAU

```

630      .SBTTL GLOBAL SUBROUTINES
631
632      :SUBROUTINE TO CHECK FOR CONTROLLER ERRORS.
633      :TRANSLATE THE ERROR BITS INTO PLAIN LANGUAGE AND MERGE THEM
634      :INTO THE ERROR BUFFER (EM99) FOR SUBSEQUENT PRINTING.
635      :DRIVE ERROR IS INCLUDED, BUT IS IGNORED IN THE DISKLESS TESTS.
636
637      :CALL: JSR      R5,GETERR
638            BR      XX      ; NO ERRORS FOUND, RETURN.
639            ERRXX      ; ERROR, RETURN.
640
641      GETERR: MOV      R1,-(SP)      ; SAVE R1.
642            CLR      DERFLG      ; CLEAR OUT DRIVE ERROR FLAG
643            MOV      #EM99,R1     ; SET TEXT BUFFER POINTER.
644            BIT      #176000,E.CS ; ANY ERRORS THERE ??
645            BNE     1$           ; BR IF SO.
646            JSR     R5,FIX       ; NO, JUST SAY 'NONE'...
647            NON
648            BR      10$         ;...AND RETURN.
649
650      1$:  TST      E.CS          ; IS COMPOSITE ERROR SET ? (BETTER BE)
651            BPL     2$           ; IF NOT SOMETHING'S WRONG !!!
652            JSR     R5,FIX       ; YES, PUT "CERR" IN STRING.
653            CERR
654      2$:  BIT      #DERR,E.CS    ; DRIVE ERROR SET?
655            BEQ     3$           ; BR IF NOT.
656            INC     DERFLG      ; YES, SET DRV ERROR FLAG
657            JSR     R5,FIX       ; ADD "DRV" TO STRING.
658            DEMES
659            BIT      #036000,E.CS ; ANY OTHERS ???
660            BEQ     10$         ; NO, IGNORE THE DRIVE ERROR (DISKLESS).
661            TST     (R5)+        ; YES, BUMP PC TO TAKE ERROR RETURN.
662            BIT     #NXM!PAR,E.CS ; NON-EX OR PARITY ??
663            BEQ     5$           ; BR IF NOT.
664            MOV     #NXMMES,4$
665            BIT     #OPI,E.CS
666            BEQ     +10
667            MOV     #PARMES,4$
668            JSR     R5,FIX       ; ADD "NXM" OR "PAR".
669            NXMMES
670      4$:  BIT     #DLT!HNF,E.CS  ; DATA LATE OR HEADER NOT FOUND ??
671            BEQ     7$           ; BR IF NOT.
672            MOV     #DLTMES,6$
673            BIT     #OPI,E.CS
674            BEQ     +10
675            MOV     #HNFMES,6$
676            JSR     R5,FIX       ; ADD "DLT" OR "HNF".
677            DLTMES
678      6$:  DLTMES
679      7$:  BIT     #DCRC!HCRC,E.CS ; EITHER CRC ??
680            BEQ     9$           ; BR IF NOT.
681            MOV     #DCRCMES,8$
682            BIT     #OPI,E.CS
683            BEQ     +10
684            MOV     #HCRCMES,8$
685            JSR     R5,FIX       ; ADD "DCRC" OR "HCRC".
686            HCRCMES
687      8$:  DCRCMES

```



```

686 007760 032737 002000 003364 9$: BIT #OPI,E.CS ; 'OPI' SET ??
687 007766 001403 BEQ 10$
688 007770 004537 010004 JSR R5,FIX ; ADD 'OPI' TO STRING.
689 007774 030623 OPIMES
690 007776 105011 10$: CLRB (R1) ; TERMINATE ERROR LIST...
691 010000 012601 MOV (SP)+,R1 ; ...AND RESTORE R1.
692 010002 000205 RTS R5 ; RETURN.
693
694 ;ROUTINE TO MOVE ASCII STRINGS
695 ;USES REGISTERS R1 - WHERE STRING IS BEING BUILT
696
697 ;
698 CALL JSR R5,FIX ;ADDRESS OF STRING TO MOVE
699 ;.WORD
700 010004 012500 FIX: MOV (R5)+,R0 ;GET ADDRESS AND MOVE RETURN
701 010006 112021 1$: MOV (R0)+,(R1)+ ;GET BYTE AND UPDATE
702 010010 001376 BNE 1$ ;WATCH 0 BYTE TERMINATOR
703 010012 105741 TSTB -(R1) ;BACK UP OVER ZERO BYTE
704 010014 000205 RTS R5 ;EXIT
705
706 ;LOAD REGISTERS BEFORE FUNCTION
707 ;CALL: JSR R5,BEFORE
708
709 010016 017737 173302 003350 BEFORE: MOV @RLCS,B.CS ;READ CS...
710 010024 042737 176000 003350 BIC #176000,B.CS ;...ERRORS WILL BE CLEARED ON XCT.
711 010032 017737 173270 003352 MOV @RLBA,B.BA ;READ BA
712 010040 017737 173264 003354 MOV @RLDA,B.DA ;READ DA
713 010046 013737 003436 003356 MOV INIMP,B.MP ; INITIAL MP IS THE WORD COUNT.
714 010054 023727 003402 000003 CMP RLTP,#RLV12X
715 010062 001003 BNE 1$
716 010064 017737 173244 003360 1$: MOV @RLBAE,B.BAE ; READ BAE IF ENABLED.
717 010072 000205 RTS R5
718
719 ;LOAD REGISTERS AT ERROR OR DONE.
720 ;CALL: JSR R5,AFTER
721
722 010074 017737 173224 003364 AFTER: MOV @RLCS,E.CS ;READ CS
723 010102 017737 173220 003366 MOV @RLBA,E.BA ;READ BA
724 010110 017737 173214 003370 MOV @RLDA,E.DA ;READ DA
725 010116 017737 173210 003372 MOV @RLMP,E.MP ;READ MP
726 010124 005737 036704 TST MIMIC ; ***** IF MIMIC
727 010130 001003 BNE 1$ ; ***** BYPASS 2ND MP READ
728 010132 017737 173174 003374 1$: MOV @RLMP,E.MP1 ;READ MP (MAINT 2ND CRC CHECK ONLY).
729 010140 023727 003402 000003 CMP RLTP,#RLV12X
730 010146 001003 BNE 2$
731 010150 017737 173160 003376 2$: MOV @RLBAE,E.BAE ; READ BAE IF ENABLED.
732 010156 000205 RTS R5
733
734 ;ROUTINE TO DELAY IN 1 MSEC INCREMENTS.
735 ;CALIBRATED FOR LSI'S ONLY.
736 ;DELAY COUNT IS 291. (443B) FOR LSI 11/23.
737 ; 120. (170B) FOR LSI 11 OR 11/2.
738 ; 500. (764B) IF CPU TYPE IS UNKNOWN.
739
740 ;CALL: JSR R5,WDELAY
741 ; N ; DELAY N MSECS.

```

```

742
743 010160 010146
744 010162 010246
745 010164 012502
746 010166 013701 003462
747 010172 005301
748 010174 001376
749 010176 005302
750 010200 001372
751 010202 012602
752 010204 012601
753 010206 000205
754
755
756
757
758
759
760
761
762
763
764
765 010210 012500
766 010212 012537 003436
767 010216 042700 177661
768 010222 053700 003346
769 010226 052700 000200
770 010232 010077 173066
771 010236 013777 010352 173062
772 010244 023727 003402 000003
773 010252 001413
774 010254 013700 010354
775 010260 042700 177774
776 010264 006300
777 010266 006300
778 010270 006300
779 010272 006300
780 010274 050077 173024
781 010300 000403
782 010302 013777 010354 173024
783 010310 013777 003440 173012
784 010316 013777 003436 173006
785 010324 004537 010016
786 010330 042777 000200 172766
787 010336 005737 036704
788 010342 001402
789 010344 004737 036706
790 010350 000205
791
792 010352 004364
793 010354 000000
794
795
796
797

:WDELAY: MOV R1,-(SP)
: MOV R2,-(SP)
: MOV (R5)+,R2 ;APPROX MSEC DELAY
2$: MOV DLYCNT,R1 ; GET 1 MSEC TIMER.
3$: DEC R1 ;START LOOP
: BNE 3$
: DEC R2 ;CHECK ON MSECs REQUESTED
: BNE 2$ ;BRANCH AND DO ANOTHER LOOP
4$: MOV (SP)+,R2 ;SETUP FOR RETURN AFTER DELAY
: MOV (SP)+,R1
: RTS R5

:ROUTINE TO LOAD RLCS WITH RLV11/12 MAINTENANCE FUNCTION
: EITHER FLAG DRIVEN OR INTERRUPT MODE.
:CALL: JSR R5, LDFUN
: .WORD MAINT ; OR MAINT!INTEN
: .WORD -N ; WORD COUNT.

:LOCATIONS 'BA16' AND 'BA22' COMPRISE A 22 BIT PHYSICAL BUS ADDRESS
: AND MUST BE SET BY THE CALLING ROUTINE PRIOR TO ENTRY.

LDFUN: MOV (R5)+,R0 ;BUILD THE FUNCTION IN R0.
: MOV (R5)+,INIMP ; SAVE WORD COUNT FOR LATER.
: BIC #177661,R0 ;MASK GARBAGE BITS.
: BIS DRIVE,R0 ;INSERT DRIVE (0)...
: BIS #CRDY,R0 ;...AND THE READY BITS.
: MOV R0,@RLCS ;LOAD UP THE CSR.
: MOV BA16,@RLBA ;LOAD BUS ADDRESS <15:0>.
: CMP RLTP,#RLV12X ;RLV12 WITH BAE ??
: BEQ 1$ ;BR IF SO.
: MOV BA22,R0 ;NO, GET EXTENSION IF ANY...
: BIC #^C3,R0 ;...KEEP <17:16> ONLY...
: ASL R0
: ASL R0
: ASL R0
: ASL R0 ;...MOVE 'EM TO <5:4>...
: BIS R0,@RLCS ;...AND INSERT INTO CS.
: BR 2$
1$: MOV BA22,@RLBAE ; RLV12X, SET BA EXTENSION <21:16>.
2$: MOV INIDA,@RLDA ;LOAD DISK ADDRESS.
: MOV INIMP,@RLMP ;LOAD WORD COUNT.
: JSR R5,BEFORE ;SAVE STATE BEFORE FUNCTION.
: BIC #CRDY,@RLCS ;XCT MAINT FUNCTION.
: TST MIMIC ;***** IF MIMIC
: BEQ 3$ ;*****
: JSR PC,EMURLV ;***** EMULATE THE RLV FUNCTION
3$: RTS R5 ;RETURN

BA16: BUF1 ; 16 BIT BUFFER ADDRESS <15:0>
BA22: 0 ; 6 BIT EXTENSION <21:16> IN <5:0>.

:ROUTINE TO FILL BUFFER WITH DATA PATTERNS FOR RLV MAINTENANCE.
:BUF1 IS FILLED WITH 256 WORD PATTERN (OR COMPLEMENTING PATTERN).
:BUF2 IS FILLED WITH 255 ZEROS AND 123456.

```



```

798
799
800
801
802 010356 005000
803 010360 000402
804 010362 012700 177777
805 010366 010146
806 010370 010246
807 010372 012537 003424
808 010376 012701 004364
809 010402 012702 000400
810 010406 013721 003424
811 010412 005700
812 010414 001402
813 010416 005137 003424
814 010422 005302
815 010424 001370
816 010426 012702 000377
817 010432 005021
818 010434 005302
819 010436 001375
820 010440 012721 123456
821 010444 012602
822 010446 012601
823 010450 000205
824
825
826
827
828
829
830 010452 010146
831 010454 010246
832 010456 012701 004364
833 010462 012702 000400
834 010466 004537 010526
835 010472 013721 003452
836 010476 005302
837 010500 001372
838 010502 012702 000377
839 010506 005021
840 010510 005302
841 010512 001375
842 010514 012721 123456
843 010520 012602
844 010522 012601
845 010524 000205
846
847
848
849
850
851
852
853

```

```

:CALL: JSR R5,SETPAT(SETCMP)
:      .WORD N ;PATTERN FOR BUFFER
SETPAT: CLR R0 ; SET NO COMPLIMENT...
        BR +6 ;...AND SKIP NEXT.
SETCMP: MOV #-1,R0 ; SET COMPLIMENT.
        MOV R1,-(SP)
        MOV R2,-(SP)
        MOV (R5)+,TMP0 ; DATA TO STORE.
        MOV #BUF1,R1 ; BUFFER POINTER.
        MOV #256,R2 ; WORD COUNT
1$: MOV TMP0,(R1)+
        TST R0 ; ARE WE COMPLIMENTING ??
        BEQ 2$ ; NO, SKIP NEXT.
        COM TMP0 ; YES, COMPLIMENT IT.
2$: DEC R2
        BNE 1$
3$: MOV #255,R2 ; NOW CLEAR THE REST OF...
        CLR (R1)+ ;...THE BUFFER.
        DEC R2
        BNE 3$
        MOV #123456,(R1)+ ; STORE IN LAST BUFFER WORD.
        MOV (SP)+,R2
        MOV (SP)+,R1
        RTS R5

:ROUTINE TO SETUP BUFFER WITH RANDOM NUMBERS FOR RLV11 MAINT. FUNCTION
:SAME PATTERN IS USED FOR EACH CONTROLLER
:END OF PASS WILL CHANGE RANDOM PATTERN PRIMES
:CALL JSR R5,SETRAN
SETRAN: MOV R1,-(SP)
        MOV R2,-(SP)
        MOV #BUF1,R1 ; FIRST BUFFER START
        MOV #256,R2 ; BUFFER COUNT
1$: JSR R5,RAND ; GET RANDOM NUMBER
        MOV HINUM,(R1)+ ; STORE IN BUFFER
        DEC R2
        BNE 1$
2$: MOV #255,R2 ; NOW CLEAR THE REST.
        CLR (R1)+
        DEC R2
        BNE 2$
        MOV #123456,(R1)+ ; STORE IN LAST BUFFER WORD
        MOV (SP)+,R2
        MOV (SP)+,R1
        RTS R5

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

```



```

854 010526 010146 RAND: MOV R1,-(SP) :PUSH R1 ON STACK
855 010530 010246 MOV R2,-(SP) :PUSH R2 ON STACK
856 010532 010346 MOV R3,-(SP) :PUSH R3 ON STACK
857 010534 013703 003454 MOV LONUM,R3 :SET R3 WITH LOW
858 010540 013701 003452 MOV HINUM,R1 :SET R1 WITH HIGH
859 010544 012702 177771 MOV #-7,R2 :SET SHIFT COUNTER
860 010550 006303 1$: ASL R3 :SHIFT R3 LEFT AND
861 010552 006101 ROL R1 :ROTATE CARRY INTO R1 AND
862 010554 005202 INC R2 :CHECK FOR DONE
863 010556 001374 BNE 1$ :CONTINUE SHIFT LOOP
864 010560 063703 003454 ADD LONUM,R3 :ADD NUMBER TO MAKE X 129
865 010564 005501 ADC R1 :PROPOGATE CARRY
866 010566 063701 003452 ADD HINUM,R1 :ADD NUMBER TO MAKE X 129
867 010572 062703 001057 ADD #1057,R3 :ADD LOW CONSTANT
868 010576 005501 ADC R1 :PROPOGATE CARRY
869 010600 062701 047401 ADD #47401,R1 :ADD HIGH CONSTANT
870 010604 010337 003454 MOV R3,LONUM :SAVE R3
871 010610 010137 003452 MOV R1,HINUM :SAVE R1
872 010614 012603 MOV (SP)+,R3 :POP STACK INTO R3
873 010616 012602 MOV (SP)+,R2 :POP STACK INTO R2
874 010620 012601 MOV (SP)+,R1 :POP STACK INTO R1
875 010622 000205 RTS R5 :RETURN
876
877 :
878 : ROUTINES TO SET A FLAG ON BUS-ERROR AND/OR RL INTERRUPT.
879 010624 005227 TRAP4: INC (PC)+
880 010626 000000 TRPFLG: 0
881 010630 000002 RTI
882
883 010632 005227 RLINT: INC (PC)+
884 010634 000000 INTFLG: 0
885 010636 000002 CLKRTI: RTI : USE TO DISMISS FREE-RUNNING CLOCK.
886
887 :
888 : ROUTINE TO WAIT FOR CONTROLLER READY
889 : THE 800 MSEC INTERVAL APPLIES TO LSI'S ONLY !!!
890 : IN ANY CASE, IT'S JUST A KEEP-ALIVE TIMER.
891 : ON EXIT, -4(SP) HOLDS THE REMNANTS OF THE 800. MS COUNT.
892 010640 012746 001440 WTCRDY: MOV #800,-(SP) :SET 800 MSEC TIMER.
893 010644 032777 000200 172452 1$: BIT %CRDY,%RLCS :CONTROLLER READY ??
894 010652 001014 BNE 2$ :YES, EXIT
895 010654 004537 010160 JSR R5,WDELAY :WAIT A WHILE
896 010660 000001 1 :APPROX A MILLISECOND
897 010662 005316 DEC (SP) :CHECK IF TIME UP
898 010664 001367 BNE 1$ :NO GO BACK
899 010666 004537 010074 JSR R5,AFTER :GET REGISTERS FOR ERROR.
900 010672 104455 DFERR CRTIM,ERRO :CONTROLLER TIMED OUT
(5) 010672 104455
(6) 010674 000144 TRAP CSERDF
(6) 010676 030476 .WORD 100
(6) 010700 026164 .WORD CRTIM
901 010702 000402 .WORD ERRO
902 010704 004537 010074 2$: SKP2
903 010710 005726 JSR R5,AFTER :GET REGISTERS
904 010712 000205 TST (SP)+ :CLEAN THE STACK...
905 : RTS R5 :...AND RETURN.

```

```

906 :RLV11 MAINTENANCE SUBROUTINE FOR CRC CALCULATIONS
907 :ROUTINE TO RETRIEVE PATTERN AND CALCULATE CRC OF PATTERN+3
908 :AND CRC OF CRC OF PATTERN+4.
909 :PATTERN IS SAVED IN "INIDA" FOR SUBSEQUENT LOADING FOR DA.
910 :CRC OF PATTERN+3 WILL BE STORED IN "GDCRC3".
911 :CRC OF CRC OF PATTERN+4 WILL BE STORED IN "GDCRC4".
912
913 :CALL: JSR R5,CALCRC
914 :      .WORD N ;PATTERN FOR INITIAL DA
915
916 010714 012537 003440 :CALCRC: MOV (R5)+,INIDA ;STORE PATTERN
917 010720 013737 003440 003422 MOV INIDA,TEMP1
918 010726 113737 003422 003420 MOVB TEMP1,TEMP5
919 010734 062737 000003 003420 ADD #3,TEMP5 ;ADD 3 TO PATTERN
920 010742 113737 003420 003422 MOVB TEMP5,TEMP1
921 010750 013737 003422 010764 MOV TEMP1,1$
922 010756 004537 011060 JSR R5,SIMBCC ;CALCULATE EXPECTED CRC
923 010762 000020 16. ;DATA BITS
924 010764 000000 1$: .WORD 0 ;INITIAL PATTERN+3
925 010766 000000 .WORD 0
926 010770 013737 003410 003442 MOV CALBCC,GDCRC3 ;SAVE CRC OF PATTERN+3
927 010776 005237 003420 INC TEMP5 ;VALUE=PATTERN+4
928 011002 113737 003420 003422 MOVB TEMP5,TEMP1
929 011010 013737 003422 011024 MOV TEMP1,2$
930 011016 004537 011060 JSR R5,SIMBCC ;CALCULATE EXPECTED CRC
931 011022 000020 16. ;DATA BITS
932 011024 000000 2$: .WORD 0 ;INITIAL PATTERN+4
933 011026 000000 .WORD 0 ;STARTING CRC=0
934 011030 013737 003410 011044 MOV CALBCC,3$ ;STORE CRC FOR NEXT CALL
935 011036 004537 011060 JSR R5,SIMBCC ;CAL. CRC OF CRC OF DA+4
936 011042 000020 16. ;DATA BITS
937 011044 000000 3$: .WORD 0 ;CRC OF DA+4
938 011046 000000 .WORD 0 ;STARTING CRC=0
939 011050 013737 003410 003444 MOV CALBCC,GDCRC4 ;SAVE CRC OF CRC OF DA+4
940 011056 000205 RTS R5
941
942 :SUBROUTINE TO CALCULATE A CRC.
943 :ROUTINE WILL CALCULATE A CRC-16 CRC ON A WORD OF
944 :1-16 BITS IN LENGTH, RESULT IS RETURNED IN "CALBCC"
945
946 :CALL: JSR R5,SIMBCC
947 :      .WORD ;NUMBER OF BITS (1-16)
948 :      .WORD ;DATA FOR CRC CALCULATION
949 :      .WORD ;PREVIOUS OR STARTING CRC
950 :      (SHOULD BE ZEROED FOR START)
951
952 011060 010046 :SIMBCC: MOV R0,-(SP) ;SAVE R0
953 011062 010146 MOV R1,-(SP) ;SAVE R1
954 011064 010246 MOV R2,-(SP) ;SAVE R2
955 011066 012537 003412 MOV (R5)+,TEMP2 ;GET NUMBER OF BITS
956 011072 012537 003414 MOV (R5)+,TEMP3 ;GET DATA FOR CRC CALCULATION
957 011076 012537 003416 MOV (R5)+,TEMP4 ;GET STARTING CRC
958 011102 005037 003406 1$: CLR BCCFBK ;
959 011106 013700 003416 MOV TEMP4,R0 ;GET PRESENT CRC
960 011112 006037 003414 ROR TEMP3 ;ROTATE NEW DATA
961 011116 005500 ADC R0 ;MERGE NEW WITH OLD

```

962	011120	032700	000001		BIT	#1,R0		:BIT 0 SET
963	011124	001402			BEQ	2\$:IF NOT CONTINUE
964	011126	005137	003406		COM	BCCFBK		:
965	011132	013700	003404	2\$:	MOV	XPOLY,R0		:GET CRC POLYNOMIAL (CRC-16)
966	011136	005100			COM	R0		:COMPLIMENT POLYNOMIAL
967	011140	040037	003406		BIC	R0,BCCFBK		
968	011144	000241			CLC			:CLEAR CARRY
969	011146	006037	003416		ROR	TEMP4		
970	011152	013700	003406		MOV	BCCFBK,R0		
971	011156	013701	003416		MOV	TEMP4,R1		
972	011162	010102			MOV	R1,R2		
973	011164	040100			BIC	R1,R0		
974	011166	043702	003406		BIC	BCCFBK,R2		
975	011172	050200			BIS	R2,R0		
976	011174	043737	003404	003416	BIC	XPOLY,TEMP4		
977	011202	050037	003416		BIS	R0,TEMP4		
978	011206	005337	003412		DEC	TEMP2		
979	011212	001333			BNE	1\$		
980	011214	013737	003416	003410	MOV	TEMP4,CALBCC		
981	011222	012602			MOV	(SP)+,R2		
982	011224	012601			MOV	(SP)+,R1		
983	011226	012600			MOV	(SP)+,R0		
984	011230	000205			RTS	R5		:RETURN

986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041

011232 000000
011234 000000
011236 000000
011240 000020

177572
177574
177576
172516
172340
172300

011242 012737 000020 011240
011250 013746 000004
011254 013746 000006
011260 013746 000114
011264 013746 000116
011270 010605

011272 012737 000114 000114
011300 012737 000002 000116
011306 012737 011332 000004
011314 012737 000340 000006
011322 005737 177572
011326 000240
011330 000417

011332 012737 011360 000004
011340 005002
011342 005003
011344 005712
011346 000240
011350 062702 004000
011354 005203
011356 000772
011360 162702 000002
011364 005001
011366 000506

```
.SBTTL MEMORY SIZER
*****
ROUTINE TO SIZE AVAILABLE MEMORY.
FREELY ADAPTED FROM "SSIZE" IN SYSMAC.SML (C3).
THIS ROUTINE MUST RESIDE WITHIN THE FIRST 24K (0-137776).

USE KT IF IT'S AVAILABLE, OTHERWISE SIZE UP TO 30K IN THE
TRADITIONAL FASHION. RETURN WITH:

.MSIZE: .WORD 0      ; TOTAL MEMORY SIZE (K WORDS)
.LSTPG: .WORD 0      ; PAGE ADDRESS (PAF) OF LAST 1K PAGE...
                        ; ...OR ZERO IF KT NOT AVAILABLE.
.LSTAD: .WORD 0      ; LAST VIRTUAL ADDRESS IN LAST PAGE...
                        ; ...OR LAST ADDRESS UNDER 30K (IF NO KT).
.ABUSW: .WORD 16.    ; ADDRESS BUS WIDTH, 16, 18, OR 22.

ALL GENERAL REGISTERS ARE USED BUT NOT SAVED.
MEMORY PARITY ERRORS (IF ANY) ARE IGNORED.

MMR0= 177572          ; KT CONTROL REGISTERS.
MMR1= 177574
MMR2= 177576
MMR3= 172516
KIPAR0= 172340       ; KERNAL, I SPACE, PAR 0.
KIPDR0= KIPAR0-40   ; PDR 0.

*****
.SIZE:  MOV #16,ABUSW ; ASSUME 16 BIT ADDRESSING.
        MOV @#4,-(SP) ; SAVE BUS-ERROR...
        MOV @#6,-(SP)
        MOV @#114,-(SP) ; ...AND PARITY VECTORS.
        MOV @#116,-(SP)
        MOV SP,R5      ; SAVE STACK POINTER IN R5.

        MOV #116,@#114 ; IGNORE PARITY ERRORS.
        MOV #RTI,@#116
        MOV #18,@#4    ; SIZE USING '18' IF KT ISN'T THERE.
        MOV #340,@#6
        TST MMR0
        BR 240

        .SIZKT        ; OTHERWISE, SIZE USING THE KT.

1$:    MOV #38,@#4    ; NO KT -- SET TRAP CATCHER.
        CLR R2        ; FIRST ADDRESS (0).
        CLR R3
        TST (R2)     ; SIZE FROM 0 UP...
        BR 240
        ADD #4000,R2 ; ...IN 1K STEPS...
        INC R3
        BR 2$
        BR 2$        ; ...UNTIL WE TRAP.
3$:    SUB #2,R2     ; R2 = LAST VIRTUAL ADDRESS.
        CLR R1      ; R1 = 0 (PAF DOESN'T APPLY).
        BR .SIZXIT  ; RETURN.
```

1042	011370	012701	172340			.SIZKT: MOV	#KIPAR0,R1	:	1ST 'PAR' ADDRESS...
1043	011374	012702	172300			MOV	#KIPDR0,R2	:	...AND IT'S 'PDR'...
1044	011400	012703	000010			MOV	#8.,R3	:	...AND THERE ARE 8 OF EACH.
1045	011404	005000				CLR	R0	:	1ST PAGE IS ZERO.
1046	011406	010021			1\$:	MOV	R0,(R1)+	:	SET PAR'S = 0, 4K, 8K ... 28K.
1047	011410	012722	077406			MOV	#77406,(R2)+	:	SET PDR'S = 4K, EX-UP, READ/WRITE.
1048	011414	062700	000200			ADD	#200,R0	:	SET NEXT PAGE PAF (+4K)...
1049	011420	077306				SOB	R3,1\$:	...AND LOOP UNTIL ALL LOADED.
1050	011422	012741	177600			MOV	#177600,-(R1)	:	PAR7 IS THE I/O PAGE (PAF).
1051	011426	005041				CLR	-(R1)	:	PAR6 WILL DO THE SIZING.
1052	011430	005003				CLR	R3	:	R3 WILL COUNT THE K'S.
1053									
1054	011432	012737	011504	000004		MOV	#2\$,@#4	:	TRAP TO 2\$ IF NO 22 BIT SUPPORT.
1055	011440	005737	172516			TST	MMR3	:	22 BITS SUPPORTED ??
1056	011444	012737	000020	172516		MOV	#20,MMR3	:	MUST BE, SET 22 BIT MODE...
1057	011452	012737	011520	000004		MOV	#3\$,@#4	:	...TRAP TO 3\$ IF 22 BIT ADDRESS IS NXM.
1058	011460	005237	177572			INC	MMR0	:	***** KT ON *****
1059	011464	012711	010000			MOV	#10000,(R1)	:	SET PAR6 AT START OF 22 BIT LAND...
1060	011470	023737	000200	140200		CMP	@#200,140200	:	...AND LOOK FOR WRAP-AROUND.
1061	011476	001010				BNE	3\$:	WE'RE HERE IF IT DIDN'T TRAP...
1062								:	...BRANCH IF IT DIDN'T WRAP-AROUND.
1063	011500	005037	172516			CLR	MMR3	:	WRAP-AROUND -- MUST BE 18 BITS ONLY.
1064	011504	012737	000022	011240	2\$:	MOV	#18.,ABUSW	:	SET THE BUS WIDTH = 18.
1065	011512	012704	007600			MOV	#7600,R4	:	SET SIZER LIMIT = 124K.
1066	011516	000405				BR	4\$:	
1067	011520	012737	000026	011240	3\$:	MOV	#22.,ABUSW	:	WE HAVE A REAL 22 BIT ADDRESS SPACE.
1068	011526	012704	170000			MOV	#170000,R4	:	SET SIZER LIMIT = 1920K.
1069	011532				4\$:			:	
1070	011532	012737	011560	000004		.SIZMEM: MOV	#2\$,@#4	:	TRAP TO 2\$ WHEN DONE SIZING.
1071	011540	005011				CLR	(R1)	:	SET PAR6 AT 1ST PAGE (0).
1072	011542	005737	140000		1\$:	TST	140000	:	SIZE USING PAR6 (+0)...
1073	011546	062711	000040			ADD	#40,(R1)	:	...IN 1K STEPS.
1074	011552	005203				INC	R3	:	
1075	011554	021104				CMP	(R1),R4	:	REACHED LIMIT ??
1076	011556	103771				BLO	1\$:	LOOP IF NOT.
1077	011560	011100			2\$:	MOV	(R1),R0	:	DONE, SAVE FINAL PAR6...
1078	011562	012711	001400			MOV	#1400,(R1)	:	...AND RESET IT TO BANK 6 (24K).
1079	011566	005037	177572			CLR	MMR0	:	***** KT OFF *****
1080	011572	010001				MOV	R0,R1	:	RECOVER SIZING RESULT.
1081	011574	162701	000040			SUB	#40,R1	:	R1 = LAST 1K PAGE (PAF).
1082	011600	012702	003776			MOV	#3776,R2	:	R2 = LAST ADDR IN THAT PAGE.
1083								:	
1084	011604	010137	011234			.SIZXIT: MOV	R1,.LSTPG	:	RETURN LAST PAGE (PAF)...
1085	011610	010237	011236			MOV	R2,.LSTAD	:	...LAST VIRTUAL ADDRESS...
1086	011614	010337	011232			MOV	R3,.MSIZE	:	...AND TOTAL MEMORY SIZE (K).
1087	011620	010506				MOV	R5,SP	:	RECOVER OUR STACK POINTER...
1088	011622	012637	000116			MOV	(SP)+,@#116	:	...AND THE ERROR VECTORS.
1089	011626	012637	000114			MOV	(SP)+,@#114	:	
1090	011632	012637	000006			MOV	(SP)+,@#6	:	
1091	011636	012637	000004			MOV	(SP)+,@#4	:	
1092	011642	000207				RTS	PC	:	...AND RETURN.

1094
1095
1096
1097
1098
1099
(3)
1100
1101
1102
1103
1104
(8)
(7)
(6)
(3)
(4)
(4)
1105
(8)
(7)
(6)
(3)
(4)
(4)
1106
(8)
(7)
(6)
(3)
(4)
(4)
1107
(4)
(3)
1108
1109
1110
1111
1112
1113
(3)
(3)

011644
011644
011644 105037 012104
011650 005737 011234
011654 001003
011656 112737 000040 012104
011664
011664 012746 012104
011670 012746 011764
011674 012746 000002
011700 010600
011702 104417
011704 062706 000006
011710
011710 013746 011232
011714 012746 012014
011720 012746 000002
011724 010600
011726 104417
011730 062706 000006
011734
011734 013746 011240
011740 012746 012047
011744 012746 000002
011750 010600
011752 104417
011754 062706 000006
011760
011760 000167
011762 000126
011764 047045 040445 020040
012014 047045 040445 020040
012047 045 022516 020101
012104 047040 052117 000
012112
012112
012112 104425

SBTTL REPORT ENVIRONMENT

SUBROUTINE TO DISPLAY THE CURRENT OPERATING ENVIRONMENT.
CALLED ON EVERY START/RESTART, OR DIRECTLY VIA 'PRI'.

BGNRPT

1S: CLRB MSGNOT
TST LSTPG
BNE 2S
2S: MOVB #40,MSGNOT
PRINTF #MSG2,#MSGNOT ; MMU

LSRPT::

PRINTF #MSG3,.MSIZE ; MEM SIZE.

MOV #MSGNOT,-(SP)
MOV #MSG2,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #6,SP

MOV .MSIZE,-(SP)
MOV #MSG3,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #6,SP

3S: PRINTF #MSG4,.ABUSW ; ADDRESS SPACE.

MOV .ABUSW,-(SP)
MOV #MSG4,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C\$PNTF
ADD #6,SP

EXIT RPT

.WORD JSJMP
.WORD L10012-2-

MSG2: .ASCIZ 'XNZA MMUXTZA AVAILABLE'
MSG3: .ASCIZ 'XNZA MEMORY SIZE XD4ZA KW'
MSG4: .ASCIZ 'XNZA XD2ZA BIT ADDRESSINGXN'
MSGNOT: .ASCIZ ' NOT'
.EVEN
ENDRPT

L10012: TRAP CSRPT


```
1115 .SBTTL
1116 .SBTTL RL DISKLESS CONTROLLER TESTS.
1117 .SBTTL
1118 .SBTTL 1 -- RLCS ADDRESSABILITY.
1119
1120 012114 STARS
(2) :*****
1121 :TEST TO SEE IF WE CAN ADDRESS THE CONTROL
1122 :AND STATUS REGISTER. IF WE TRAP WE WILL REPORT
1123 :THE ERROR AND ABORT. AFTER THIS TEST WE ONLY KNOW
1124 :THAT WE CAN ADDRESS THE REGISTER.
1125 012114 STARS
(2) :*****
1126 012114 BEGIN.TEST
(4) 012114
1127 012114 005037 010626 1S: CLR TRPFLG ;CLEAR TRAP OCCURANCE T1::
1128 012120 012777 177777 171176 MOV #177777,@RLCS ; WRITE RLCS.
1129 012126 000240 000240 240,240
1130 012132 005777 171166 TST @RLCS ; READ RLCS.
1131 012136 000240 000240 240,240
1132 012142 005737 010626 TST TRPFLG ; EITHER ONE TRAP ??
1133 012146 001413 BEQ 3$ ; NO, PROCEED.
1134 012150 013737 003324 003432 MOV RLCS,GDDAT ;SET UP ERROR DATA
1135 012156 SFERR EM1,ERR1 ;BUS TIMEOUT IN ADDRESSING RLCS
(5) 012156 104454 TRAP CSERSF
(6) 012160 000002 .WORD 2
(6) 012162 030706 .WORD EM1
(6) 012164 026170 .WORD ERR1
1136 012166 DODU UNITST ; DROP...
(3) 012166 013700 003322 MOV UNITST,RC
(3) 012172 104451 TRAP CSDODU
1137 012174 DOCLN ;...AND ABORT. TRAP CSDCLN
(3) 012174 104444
1138 012176 3$: ENDTST L10013: TRAP CSETST
(3) 012176
(3) 012176 104401
```

1140
1141
1142 012200
(2)
1143
1144
1145
1146
1147 012200
(2)
1148 012200
(4)
1149 012200 005037 010626
1150 012204 012777 177777 171114
1151 012212 000240 000240
1152 012216 005777 171104
1153 012222 000240 000240
1154 012226 005737 010626
1155 012232 001413
1156 012234 013737 003326 003432
1157 012242
(5) 012242 104454
(6) 012244 000003
(6) 012246 030733
(6) 012250 026170
1158 012252
(3) 012252 013700 003322
(3) 012256 104451
1159 012260
(3) 012260 104444
1160 012262
(3) 012262
(3) 012262 104401

.SBTTL 2 -- RLBA ADDRESSABILITY.

STARS
:*****
:TEST TO SEE IF WE CAN ADDRESS THE BUS 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
:*****

BEGIN.TEST

1\$: CLR TRPFLG ;CLEAR TRAP OCCURANCE T2::
MOV #177777,@RLBA ;WRITE RLBA.
240,240
TST @RLBA ;READ RLBA.
240,240
TST TRPFLG ;ANY TRAPS ??
BEQ 3\$;NO, CONTINUE
MOV RLBA,GDDAT ;SETUP ERROR DATA
SFERR EM2,ERR1 ;BUS TIMEOUT IN ADDRESSING RLBA

DODU UNITST ;DROP...
TRAP CSERSF
.WORD 3
.WORD EM2
.WORD ERR1

DOCLN ;...AND ABORT.
MOV UNITST,RO
TRAP CSDODU

3\$: ENDTST
TRAP CSDCLN
L10014:
TRAP CSETST

1162
1163
1164 012264
(2)
1165
1166
1167
1168
1169 012264
(2)
1170 012264
(4)
1171 012264 005037 010626
1172 012270 012777 177777 171032
1173 012276 000240 000240
1174 012302 005777 171022
1175 012306 000240 000240
1176 012312 005737 010626
1177 012316 001413
1178 012320 013737 003330 003432
1179 012326
(5) 012326 104454
(6) 012330 000004
(6) 012332 030760
(6) 012334 026170
1180 012336
(3) 012336 013700 003322
(3) 012342 104451
1181 012344
(3) 012344 104444
1182 012346
(3) 012346
(3) 012346 104401

.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

:*****
:BEGIN.TEST

1S: CLR TRPFLG ;CLEAR TRAP OCCURANCE T3::
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.

3S: ENDTST
L10015: TRAP CSETST

TRAP CSERSF
.WORD 4
.WORD EM3
.WORD ERR1
MOV UNITST,R0
TRAP CSDODU
TRAP CSDCLN
TRAP CSETST

1184
1185
1186 012350
(2)
1187
1188
1189
1190
1191 012350
(2)
1192 012350
(4)
1193 012350 005037 010626
1194 012354 012777 177777 170750
1195 012362 000240 000240
1196 012366 005777 170740
1197 012372 000240 000240
1198 012376 005737 010626
1199 012402 001413
1200 012404 013737 003332 003432
1201 012412
(5) 012412 104454
(6) 012414 000005
(6) 012416 031005
(6) 012420 026170
1202 012422
(3) 012422 013700 003322
(3) 012426 104451
1203 012430
(3) 012430 104444
1204 012432
(3) 012432
(3) 012432 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

DODU UNITST ; DROP...

DOCLN ;...AND ABORT.

3\$: ENDTST

L10016:

TRAP CSERSF
.WORD 5
.WORD EM4
.WORD ERR1
MOV UNITST,RO
TRAP CSDODU
TRAP CSDCLN
TRAP CSETST

```
.SBTTL 5 -- RLBAE ADDRESSABILITY (RLV12 ONLY).  
STARS  
:*****  
: THE DEFAULT (SHIP) CONFIGURATION REQUIRES THAT THE  
: BAE REGISTER BE ENABLED. CHECK IT AND REPORT ACCORDINGLY.  
STARS  
:*****  
BEGIN.TEST  
T5::  
1206  
1207  
1208 012434  
(2)  
1209  
1210  
1211 012434  
(2)  
1212 012434  
(4) 012434  
1213 012434 023727 003402 000002 CMP RLTP,#RLV12 ; RLV12 ??  
1214 012442 103444 4$ ; NO, SKIP THIS TEST.  
1215 012444 005037 010626 1$: CLR TRPFLG ; CLEAR TRAP OCCURANCE  
1216 012450 013737 003334 003432 MOV RLBAE,GDDAT ; SET ADDRESS IN CASE OF ERROR.  
1217 012456 012777 177777 170650 MOV #-1,@RLBAE ; WRITE BAE.  
1218 012464 000240 000240 240,240 TST @RLBAE ; READ BAE.  
1219 012470 005777 170640 240,240  
1220 012474 000240 000240  
1221 012500 023727 003402 000003 CMP RLTP,#RLV12X ; DEFAULT CONFIG (W/BAE) ??  
1222 012506 001013 3$ BNE ; BR IF NOT.  
1223  
1224 012510 005737 010626 2$: TST TRPFLG ; BAE SHOULD HAVE ANSWERED -- DID IT ??  
1225 012514 001417 4$ BEQ ; BR IF SO.  
1226 012516 SFERR EM4A,ERR1 ;BUS TIMEOUT IN ADDRESSING RLBAE  
(5) 012516 104454 TRAP CSERSF  
(6) 012520 000006 .WORD 6  
(6) 012522 031032 .WORD EM4A  
(6) 012524 026170 .WORD ERR1  
1227 012526 DODU UNITST ; DROP...  
(3) 012526 013700 003322 MOV UNITST,R0  
(3) 012532 104451 TRAP CSDODU  
1228 012534 DOCLN ;...AND ABORT.  
(5) 012534 104444 TRAP CSDCLN  
1229  
1230 012536 005737 010626 3$: TST TRPFLG ; BAE SHOULD HAVE TRAPPED -- DID IT ??  
1231 012542 001004 4$ BNE ; BR IF SO.  
1232 012544 DFERR EM4B,ERR1 ; BAE IS NOT DISABLED.  
(5) 012544 104455 TRAP CSERDF  
(6) 012546 000145 .WORD 101  
(6) 012550 031060 .WORD EM4B  
(6) 012552 026170 .WORD ERR1  
1233 012554 4$: ENDTST  
(3) 012554  
(3) 012554 104401 L10017: TRAP CSETST
```

1235
1236
1237
(2)
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
(2)
1248
(4)
1249
(3)
(3)
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
(5)
(6)
(6)
(6)
(6)
(6)
(6)
1274
1275
1276
(5)

012556

012556
012556
012556
012562
012564
012572
012600
012606
012610
012616
012624
012632
012640
012642
012650
012654
012656
012660
012664
012666
012674
012702
012710
012712
012714
012716
012720
012722
012726
012734
012736
012738
012740
012742
012744
012746
012754
012756
012756

012700 000340
104441
000377 170532
000200 003432
040000 170516
001403
140000 003432
177777 170502
177777 170476
023727 000003
001003
177777 170464
004537 010160
000372
004537 010160
000372
017737 170432 003434
042737 000001 003434
023737 003434 003432
001404
104455
000146
033311
026220
005037 003432
017737 170374 003434
001404
104455
000147
033345
026220
017737 170356 003434
001404
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
:
:SETPRI #PRI07 ;PRIORITY TO SEVEN T6::
:
:MOV #377,@RLCS ;LOAD ALL RLCS LOADABLE BITS
:MOV #CRDY,GDCAT ;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.
:RESE
: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 CSERDF
:WORD 102
:WORD EM69
:WORD ERR2
:
:4$: CLR GDDAT ;EXPECT 0 IN THE OTHERS.
:MOV @RLBA,BDDAT
:BEQ 5$
:DFERR EM70,ERR2 ;BUS-INIT FAILED ON RLBA.
:
:TRAP CSERDF
:WORD 103
:WORD EM70
:WORD ERR2
:
:5$: MOV @RLDA,BDDAT
:BEQ 6$
:DFERR EM71,ERR2 ;BUS-INIT FAILED ON RLDA.
:
:TRAP CSERDF
```


(6)	012760	000150											
(6)	012762	033401								.WORD	104		
(6)	012764	026220								.WORD	EM71		
1277	012766	023727	003402	000003	6\$:	CMP	RLTYP,#RLV12X	:	RLV12X ??	.WORD	ERR2		
1278	012774	001010				BNE	7\$:	WE'RE DONE IF NOT.				
1279	012776	017737	170332	003434		MOV	@RLBAE,BDDAT						
1280	013004	001404				BEQ	7\$						
1281	013006					DFERR	EM71A,ERR2	:	BUS-INIT FAILED ON RLBAE.				
(5)	013006	104455								TRAP	C\$ERDF		
(6)	013010	000151								.WORD	105		
(6)	013012	033435								.WORD	EM71A		
(6)	013014	026220								.WORD	ERR2		
1282	013016				7\$:	ENDTST							
(3)	013016												
(3)	013016	104401								L10020:			
										TRAP	C\$ETST		

```
.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
      MOV      #CSPAT,R3          ;SET UP TABLE POINTER OF PATTERNS
      BGNSEG          ;****START OF SEGMENT****
      TRAP      CSBSEG
1286 013020
(2)
1287
1288
1289
1290
1291 013020
(2)
1292 013020
(4) 013020
1293 013020 012703 004100      MOV      #CSPAT,R3          ;SET UP TABLE POINTER OF PATTERNS
1294 013024          BGNSEG          ;****START OF SEGMENT****
(3) 013024 104404          TRAP      CSBSEG
1295 013026 011337 003432      1$: MOV      (R3),GDDAT      ;GET PATTERN INTO GDDAT
1296 013032 052737 000200 003432  BIS      #200,GDDAT      ;INSURE GO IS SET
1297 013040 013777 003432 170256  MOV      GDDAT,@RLCS     ;LOAD RLCS (CONTROL AND STATUS)
1298 013046 032777 040000 170250  BIT      #DERR,@RLCS     ;IF DRIVE ERROR PRESENT
1299 013054 001403          BEQ      2$              ;THEN EXPECT DRIVE AND
1300 013056 052737 140000 003432  BIS      #ERR!DERR,GDDAT ;COMPOSITE ERROR
1301 013064 017737 170234 003434  2$: MOV      @RLCS,BDDAT   ;READ RLCS BACK
1302 013072 042737 000001 003434  BIC      #DRDY,BDDAT     ;IGNORE DRIVE READY
1303 013100 023737 003432 003434  CMP      GDDAT,BDDAT     ;DID WE READ WHAT WE LOADED
1304 013106 001404          BEQ      3$              ;YES, THEN BRANCH
1305 013110          DFERR      EM5,ERR2 ;WRONG DATA IN RLCS
(5) 013110 104455          TRAP      CSERDF
(6) 013112 000152          .WORD    106
(6) 013114 031127          .WORD    EM5
(6) 013116 026220          .WORD    ERR2
1306 013120          3$: ESCAPE  SEG          ; ESCAPE AND LOOP (IF FLA:LOE).
(3) 013120 104410          TRAP      CSESCAPE
(3) 013122 000012          .WORD    10000$-.
1307 013124 005723          TST      (R3)+          ;BUMP FOR NEXT PATTERN
1308 013126 020327 004176  CMP      R3,#CSEND      ;CHECK FOR END
1309 013132 001335          BNE      1$              ;NOT END, LOAD NEXT PATTERN
1310 013134          ENDSEG          ;****END OF SEGMENT****
(3) 013134          10000$:
(3) 013134 104405          TRAP      CSESEG
1311 013136          ENDTST
(3) 013136          L10021:
(3) 013136 104401          TRAP      CSETST
```

```
1313 .SBTTL 8 -- READ WRITE OF RLBA.  
1314 STARS  
1315 013140 :*****  
(2) :TEST THAT WE CAN WRITE/READ BITS 15 THRU 1 OF THE  
1316 :BUS ADDRESS REGISTER. FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,  
1317 :GROWING 0 AND SHIFTING 0. BIT 0 IS ALSO LOADED BUT  
1318 :SHOULD ALWAYS COME BACK AS 0  
1319 STARS  
1320 013140 :*****  
(2) :*****  
1321 013140 BEGIN.TEST  
(4) 013140  
1322 013140 012703 003670 MOV #BEGPAT,R3 ;GET START OF PATTERN LIST T8::  
1323 013144 BGNSEG ;****START OF SEGMENT**** TRAP CSBSEG  
(3) 013144 104404  
1324 013146 011337 003432 1$: MOV (R3),GDDAT ;GET PATTERN TO SEND  
1325 013152 023727 003402 000000 CMP RLTP,#RL11 ;RL11 ??  
1326 013160 001003 BNE 2$ ;NO  
1327 013162 042737 000001 003432 BIC #BIT0,GDDAT ;YES, KEEP RLBA EVEN (UNIBUS)  
1328 013170 013777 003432 170130 2$: MOV GDDAT,@RLBA ;LOAD PATTERN TO BUS ADDRESS  
1329 013176 017737 170124 003434 MOV @RLBA,BDDAT ;READ IT BACK  
1330 013204 023737 003432 003434 CMP GDDAT,BDDAT ;IS IT CORRECT?  
1331 013212 001404 BEQ 3$ ;IF SO, BRANCH  
1332 013214 DFERR EM6,ERR2 ;DATA WRONG IN RLBA  
(5) 013214 104455 TRAP CSERDF  
(6) 013216 000153 .WORD 107  
(6) 013220 031200 .WORD EM6  
(6) 013222 026220 .WORD ERR2  
1333 013224 3$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP C$ESCAPE  
(3) 013224 104410 .WORD 10000$-  
(3) 013226 000012  
1334 013230 005723 TST (R3)+ ;BUMP FOR NEXT PATTERN  
1335 013232 020327 004076 CMP R3,#ENDPAT ;CHECK FOR END  
1336 013236 001343 BNE 1$ ; LOOP TIL DONE.  
1337 013240 ENDSEG ;****END OF SEGMENT****  
(3) 013240 10000$: TRAP C$ESEG  
(3) 013240 104405  
1338 013242 ENDTST L10022: TRAP C$SETST  
(3) 013242 104401
```



```
1340 .SBTTL 9 -- READ WRITE OF RLDA.
1341
1342 013244 STARS
(2) :*****
1343 :TEST THAT WE CAN WRITE/READ THE DISK ADDRESS REGISTER
1344 :ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
1345 :GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
1346 013244 STARS
(2) :*****
1347 013244 BEGIN.TEST
(4) 013244
1348 013244 012703 003670 MOV #BEGPAT,R3 ;SET UP POINTER TO PATTERN LIST T9::
1349 013250 BGNSEG ;****START OF SEGMENT**** TRAP CSBSEG
(3) 013250 104404
1350 013252 011337 003432 1$: MOV (R3),GDDAT ;GET PATTERN
1351 013256 013777 003432 170044 MOV GDDAT,@RLDA ;LOAD PATTERN IN DA
1352 013264 017737 170040 003434 MOV @RLDA,BDDAT ;READ PATTERN BACK
1353 013272 023737 003432 003434 CMP GDDAT,BDDAT ;IS IT CORRECT?
1354 013300 001404 BEQ 2$ ;BRANCH IF CORRECT
1355 013302 DFERR EM7,ERR2 ;WRONG DATA IN RLDA
(5) 013302 104455 TRAP CSERDF
(6) 013304 000154 .WORD 108
(6) 013306 031226 .WORD EM7
(6) 013310 026220 .WORD ERR2
1356 013312 2$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA:LOE). TRAP CSESCAPE
(3) 013312 104410 .WORD 10000$-
(3) 013314 000012
1357 013316 005723 TST (R3)+ ;BUMP POINTER
1358 013320 020327 004076 CMP R3,#ENDPAT ;AT END OF PATTERNS?
1359 013324 001352 BNE 1$ ;NO, BRANCH BACK
1360 013326 ENDSEG ;****END OF SEGMENT****
(3) 013326 10000$: TRAP CSESEG
(3) 013326 104405
1361 013330 ENDTST L10023: TRAP CSETST
(3) 013330
(3) 013330 104401
```

1363
1364
1365 013332
(2)
1366
1367
1368
1369
1370 013332
(2)
1371 013332
(4) 013332
1372 013332 023727 003402 000003
1373 013340 001035
1374 013342 012703 003670
1375 013346
(3) 013346 104404
1376 013350 011337 003432
1377 013354 013777 003432 167752
1378 013362 017737 167746 003434
1379 013370 042737 177700 003432
1380 013376 023737 003432 003434
1381 013404 001404
1382 013406
(5) 013406 104455
(6) 013410 000155
(6) 013412 031254
(6) 013414 026220
1383 013416
(3) 013416 104410
(3) 013420 000012
1384 013422 005723
1385 013424 020327 004076
1386 013430 001347
(3) 013432
(3) 013432 104405
1388 013434
(3) 013434
(3) 013434 104401

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

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****
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 EMB,ERR2 ;WRONG DATA IN RLBAE

TRAP CSBSEG

TRAP CSERDF
.WORD 109
.WORD EMB
.WORD ERR2

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

TRAP CSESCAPE
.WORD 10000\$-

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

10000\$:

TRAP CSESEG

3\$: ENDTST

L10024:

TRAP CSETST

```
1390 .SBTTL 11 -- BIS AND BIC OF RLCS.
1391
1392 013436
1393 (2)
1394
1395 STARS
1396 :*****
1397 :TEST THAT WE CAN USE THE 'BIS' AND 'BIC' INSTRUCTIONS ON THE CONTROL
1398 :AND STATUS REGISTER. BITS 8,9 AND 6-1 ARE TESTED TO
1399 :SET INDIVIDUALLY AS WELL AS COLLECTIVELY WITHOUT DESTROYING
1400 :ANY PREVIOUS DATA PATTERN
1401 STARS
1402 :*****
1403 BEGIN.TEST
1404
1405 T11::
1406 MOV #CSPAT,R3 ;GET BEGINNING OF LIST
1407 BGNSEG ;**** START SEGMENT ****
1408 TRAP CSBSEG
1409
1410 104404
1411 012703 004100 167652 1$: MOV #CRDY,@RLCS ;INSURE GO IS THERE
1412 (R3),GDDAT ;SET UP EXPECTED RLCS...
1413 012777 000200 167652 1$: MOV #CRDY,GDDAT ;...IN GDDAT.
1414 011337 003432 167652 1$: BIS #CRDY,GDDAT ;*** BIT SET PATTERN IN RLCS
1415 052737 000200 003432 1$: BIS (R3),@RLCS ;IF ERROR BIT SET THEN...
1416 051377 167634 167626 1$: BIT #DERR,@RLCS ;...EXPECT IT ON THE READ BACK.
1417 032777 040000 167626 1$: BEQ 2$
1418 013476 001403 140000 003432 2$: BIS #ERR!DERR,GDDAT
1419 013500 052737 167612 003434 2$: MOV @RLCS,BDDAT ;READ RLCS TO CHECK 'BIS'
1420 013506 017737 167612 003434 2$: BIC #DRDY,BDDAT ;CLEAR OUT DRIVE READY
1421 013514 042737 000001 003434 2$: BIC #DRDY,BDDAT ;DID BIS WORK?
1422 013522 023737 003434 003432 2$: CMP BDDAT,GDDAT ;BRANCH IF OKAY
1423 013530 001404 3$: BEQ 3$ ;WRONG DATA IN RLCS ON BIS.
1424 (5) 013532 104455 TRAP CSERDF
1425 (6) 013534 000156 .WORD 110
1426 (6) 013536 032617 .WORD EM61
1427 (6) 013540 026220 .WORD ERR2
1428 1413 013542 3$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE).
1429 (3) 013542 104410 TRAP CSERDF
1430 (3) 013544 000114 .WORD 110
1431 1414 013546 012777 001776 167550 4$: MOV #1776,@RLCS ;SET ALL SETTABLE BITS
1432 013554 012737 001776 003432 4$: MOV #1776,GDDAT ;SET UP EXPECT DATA IN...
1433 013562 041337 003432 4$: BIC (R3),GDDAT ;...GDDAT
1434 013566 041377 167532 4$: BIC (R3),@RLCS ;*** XCT BIC ON RLCS.
1435 013572 032777 040000 167524 4$: BIT #DERR,@RLCS ;IF DRIVE ERROR BIT SET...
1436 013600 001403 140000 003432 4$: BEQ 5$ ;...EXPECT IT SET WHEN WE READ.
1437 013602 052737 167510 003434 5$: BIS #ERR!DERR,GDDAT
1438 013610 017737 167510 003434 5$: MOV @RLCS,BDDAT ;MOVE RLCS TO BDDAT FOR COMPARE
1439 013616 042737 000001 003434 5$: BIC #DRDY,BDDAT ;CLEAR DRIVE READY
1440 013624 023737 003434 003432 5$: CMP BDDAT,GDDAT ;DID 'BIC' WORK PROPERLY
1441 013632 001404 6$: BEQ 6$ ;BRANCH IF OKAY
1442 (5) 013634 104455 TRAP CSERDF
1443 (6) 013636 000157 .WORD 111
1444 (6) 013640 032665 .WORD EM62
1445 (6) 013642 026220 .WORD ERR2
1446 1426 013644 6$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE).
1447 (3) 013644 104410 TRAP CSERDF
1448 (3) 013646 000012 .WORD 10000S-.
1449 1427 013650 005728
1450 1428 013652 020327 004176
1451 1429 013656 001272 TST (R3)+ ;GET NEXT PATTERN
1452 CMP R3,#CSEND ;AT END OF LIST
1453 BNE 1$ ;NO, GO BACK WITH NEXT PATTERN
```


CVRLBA0 -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 H 5 PAGE 19-1
11 -- BIS AND BIC OF RLCS.

SEQ 0059

1430 013660
(3) 013660
(3) 013660 104405
1431 013662
(3) 013662
(3) 013662 104401

ENDSEG

ENDTST

100008: TRAP CSESEG

L10025: TRAP CSETST

1433
1434
1435 013664
(2)
1436
1437
1438
1439
1440 013664
(2)
1441 013664
(4) 013664
1442 013664 012703 003670
1443 013670
(3) 013670 104404
1444 013672 005077 167430
1445 013676 011337 003432
1446 013702 023727 003402 000000
1447 013710 001003
1448 013712 042737 000001 003432
1449 013720 051377 167402
1450 013724 017737 167376 003434
1451 013732 023737 003434 003432
1452 013740 001404
1453 013742
(5) 013742 104455
(6) 013744 000160
(6) 013746 032735
(6) 013750 026220
1454 013752
(3) 013752 104410
(3) 013754 000070
1455 013756 012777 177776 167342
1456 013764 012737 177776 003432
1457 013772 041337 003432
1458 013776 041377 167324
1459 014002 017737 167320 003434
1460 014010 023737 003434 003432
1461 014016 001404
1462 014020
(5) 014020 104455
(6) 014022 000161
(6) 014024 033003
(6) 014026 026220
1463 014030
(3) 014030 104410
(3) 014032 000012
1464 014034 005723
1465 014036 020327 004076
1466 014042 001313
1467 014044
(3) 014044
(3) 014044 104405
1468 014046
(3) 014046
(3) 014046 104401

.SBTTL 12 -- BIS AND BIC OF RLBA.

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

MOV #BEGPAT,R3 ;GET START OF LIST
BGNSEG ;**** START SEGMENT ****

T12::

TRAP CSBSEG

1\$:

CLR @RLBA ;CLEAR 'BA'
MOV (R3),GDDAT ;SET EXPECTED
CMP RLTP,#RL11 ; RL11 ??
BNE 2\$; NO.

:BIT 0 CAN'T SET IN RLBA (UNIBUS)
: XCT BIS RLBA WITH PATTERN
:READ 'BA'
:DID RLBA LOAD PROPERLY?
:BRANCH IF YES
:WRONG DATA IN RLBA ON BIS.

TRAP CSERDF
.WORD 112
.WORD EM63
.WORD ERR2

2\$:

BIC #1,GDDAT
BIS (R3),@RLBA
MOV @RLBA,BDDAT
CMP BDDAT,GDDAT
BEQ 3\$
DFERR EM63,ERR2

3\$:

ESCAPE SEG

: ESCAPE AND LOOP (IF FLA;LOE).

TRAP CSERDF
.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 CSERDF
.WORD 113
.WORD EM64
.WORD ERR2

5\$:

ESCAPE SEG

: ESCAPE AND LOOP (IF FLA:LOE).

TRAP CSERDF
.WORD 10000\$-

TST (R3)+ ;GET NEXT PATTERN
CMP R3,#ENDPAT ;HAVE WE COMPLETED LIST
BNE 1\$;NO, GO BACK FOR NEXT
ENDSEG

10000\$:

TRAP CSESEG

L10026:

TRAP CSETST

```

1470
1471
1472 014050
(2)
1473
1474
1475
1476 014050
(2)
1477 014050
(4) 014050
1478 014050 012703 003670
1479 014054
(3) 014054 104404
1480 014056 005077 167246
1481 014062 011337 003432
1482 014066 051377 167236
1483 014072 017737 167232 003434
1484 014100 023737 003434 003432
1485 014106 001404
1486 014110
(5) 014110 104455
(6) 014112 000162
(6) 014114 033053
(6) 014116 026220
1487 014120
(3) 014120 104410
(3) 014122 000070
1488 014124 012777 177777 167176
1489 014132 012737 177777 003432
1490 014140 041337 003432
1491 014144 041377 167160
1492 014150 017737 167154 003434
1493 014156 023737 003432 003434
1494 014164 001404
1495 014166
(5) 014166 104455
(6) 014170 000163
(6) 014172 033121
(6) 014174 026220
1496 014176
(3) 014176 104410
(3) 014200 000012
1497 014202 005723
1498 014204 020327 004076
1499 014210 001322
1500 014212
(3) 014212
(3) 014212 104405
1501 014214
(3) 014214
(3) 014214 104401

```

```

.SBTTL 13 -- BIS AND BIC OF RLDA.
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 ****
1$: CLR @RLDA ;CLEAR 'DA' TRAP CSBSEG
MOV (R3),GDDAT ;SET EXPECTED
BIS (R3),@RLDA ; XCT BIS RLDA
MOV @RLDA,BDDAT ;READ RLDA
CMP BDDAT,GDDAT ;IS RLDA CORRECT
BEQ 2$ ;IF OKAY BRANCH
DFERR EM65,ERR2 ;WRONG DATA IN RLDA ON BIS.
(5) TRAP CSERDF
(6) .WORD 114
(6) .WORD EM65
(6) .WORD ERR2
2$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP CSERDF
(3) .WORD C$ESCAPE
(3) .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.
(5) TRAP CSERDF
(6) .WORD 115
(6) .WORD EM66
(6) .WORD ERR2
4$: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP CSERDF
(3) .WORD C$ESCAPE
(3) .WORD 10000$-.
TST (R3)+ ;GET NEXT PATTERN
CMP R3,#ENDPAT ;DONE?
BNE 1$ ;NO GO BACK
ENDSEG
10000$: TRAP CSESEG
ENDTST
L10027: TRAP CSETST

```



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

1503
1504
1505 014216
(2)
1506
1507
1508
1509
1510 014216
(2)
1511 014216
(4) 014216
1512 014216 023727 003402 000003
1513 014224 001070
1514 014226 012703 003670
1515 014232
(3) 014232 104404
1516 014234 005077 167074 1S:
1517 014240 011337 003432
1518 014244 042737 177700 003432
1519 014252 051377 167056
1520 014256 017737 167052 003434
1521 014264 023737 003434 003432
1522 014272 001404
1523 014274 104455
(5) 014274 000164
(6) 014276 033171
(6) 014300 026220
1524 014304 2S:
(3) 014304 104410
(3) 014306 000076
1525 014310 012777 177777 167016 3S:
1526 014316 012737 177777 003432
1527 014324 041337 003432
1528 014330 042737 177700 003432
1529 014336 041377 166772
1530 014342 017737 166766 003434
1531 014350 023737 003434 003432
1532 014356 001404
1533 014360 104455
(5) 014360 000165
(6) 014362 033240
(6) 014364 026220
1534 014370 4S:
(3) 014370 104410
(3) 014372 000012
1535 014374 005723
1536 014376 020327 004076
1537 014402 001314
1538 014404
(3) 014404
(3) 014404 104405
1539 014406 5S:
(3) 014406
```

STARS
:*****
:TEST THAT THE 'BIS' AND 'BIC' INSTRUCTIONS WILL WORK ON THE
:BUS ADDRESS EXTENSION REGISTER. ALL BITS ARE LOADED BUT ONLY <5:0>
:ARE EXPECTED BACK. FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
:GROWING 0, AND SHIFTING 0.
STARS
:*****

BEGIN.TEST
T14::
CMP RLTP,#RLV12X ; BAE ENABLED ??
BNE 5\$; SKIP IT IF NOT.
MOV #BEGPAT,R3 ; GET START OF LIST
BGNSEG ; **** START SEGMENT **** TRAP CSBSEG
1S: CLR @RLBAE ; CLEAR 'BAE'
MOV (R3),GDDAT ; SET EXPECTED...
BIC #^C77,GDDAT ; ...ONLY <5:0> ARE VALID.
BIS (R3),@RLBAE ; XCT BIS RLBAE WITH PATTERN
MOV @RLBAE,BDDAT ; READ 'BAE'
CMP BDDAT,GDDAT ; DID IT LOAD PROPERLY?
BEQ 2\$; BRANCH IF YES
DFERR EM67,ERR2 ; WRONG DATA IN RLBAE ON BIS. TRAP CSERDF
; .WORD 116
; .WORD EM67
; .WORD ERR2
2S: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP CSESCAPE
; .WORD 10000S-.
3S: MOV #-1,@RLBAE ; SET RLBAE TO ALL 1'S
MOV #-1,GDDAT ; SET UP EXPECTED RESULTS...
BIC (R3),GDDAT ; ...IN GDDAT...
BIC #^C77,GDDAT ; ...<5:0> ONLY.
BIC (R3),@RLBAE ; XCT BIC RLBAE
MOV @RLBAE,BDDAT ; READ RLBAE
CMP BDDAT,GDDAT ; BIC WORK OKAY?
BEQ 4\$; IF YES BRANCH
DFERR EM68,ERR2 ; WRONG DATA IN RLBAE ON BIC. TRAP CSERDF
; .WORD 117
; .WORD EM68
; .WORD ERR2
4S: ESCAPE SEG ; ESCAPE AND LOOP (IF FLA;LOE). TRAP CSESCAPE
; .WORD 10000S-.
TST (R3)+ ; GET NEXT PATTERN
CMP R3,#ENDPAT ; HAVE WE COMPLETED LIST
BNE 1\$; NO, GO BACK FOR NEXT
ENDSEG
10000S: TRAP CSESEG
L10030:

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 ^{L 5} PAGE 22-1
14 -- BIS AND BIC OF RLBAE (RLV12 ONLY).

(3) 014406 104401

TRAP CSETST

SEQ 0063

1541
1542
1543 014410
(2)
1544
1545
1546
1547
1548
1549
1550
1551 014410
(2)
1552 014410
(4) 014410
1553 014410 012777 177776 166710
1554 014416 012777 177777 166704
1555 014424 023727 003402 000003
1556 014432 001003
1557 014434 012777 000077 166672
1558 014442 012777 000200 166654
1559 014450 012737 177776 003432
1560 014456 017737 166644 003434
1561 014464 023737 003432 003434
1562 014472 001405
1563 014474 104455
(5) 014474 104455
(6) 014476 000166
(6) 014500 033472
(6) 014502 026220
1564 014504
(3) 014504 104406
1565 014506 012737 177777 003432 2S:
1566 014514 017737 166610 003434
1567 014522 023737 003432 003434
1568 014530 001405
1569 014532 104455
(5) 014532 104455
(6) 014534 000167
(6) 014536 033525
(6) 014540 026220
1570 014542
(3) 014542 104406
1571 014544 023727 003402 000003 3S:
1572 014552 001016
1573 014554 012737 000074 003432
1574 014562 017737 166546 003434
1575 014570 023737 003432 003434
1576 014576 001404
1577 014600
(5) 014600 104455
(6) 014602 000170
(6) 014604 033560
(6) 014606 026220
1578 014610
(3) 014610

.SBTTL 15 -- UNIQUENESS OF RLCS.

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 T15::
MOV #-1,@RLDA ;SET RLDA TO ALL 1'S
CMP RLTP,#RLV12X
BNE 1\$
MOV #77,@RLBAE ; SET BAE TO ALL 1'S.
1\$: 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 CSERDF
.WORD 118
.WORD EM72
.WORD ERR2

CKLOOP

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

TRAP CSCLP1

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 CSERDF
.WORD 120
.WORD EM73A
.WORD ERR2

4\$: ENDTST

L10031:

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 N 5
15 -- UNIQUENESS OF RLCS. PAGE 23-1

(3) 014610 104401

TRAP CSETST

SEQ 0065

1580
1581
1582 014612
(2)
1583
1584
1585
1586
1587
1588 014612
(2)
1589 014612
(4) 014612
1590 014612 012737 000200 003432
1591 014620 032777 040000 166476
1592 014626 001403
1593 014630 052737 140000 003432
1594 014636 013777 003432 166460 1\$:
1595 014644 012777 177777 166456
1596 014652 023727 003402 000003
1597 014660 001003
1598 014662 012777 000074 166444
1599 014670 005077 166432 2\$:
1600 014674 017737 166424 003434
1601 014702 042737 000001 003434
1602 014710 023737 003432 003434
1603 014716 001405
1604 014720
(5) 014720 104455
(6) 014722 000171
(6) 014724 033617
(6) 014726 026220
1605 014730
(3) 014730 104406
1606 014732 012737 177777 003432 3\$:
1607 014740 017737 166364 003434
1608 014746 023737 003432 003434
1609 014754 001405
1610 014756
(5) 014756 104455
(6) 014760 000172
(6) 014762 033651
(6) 014764 026220
1611 014766
(3) 014766 104406
1612 014770 023727 003402 000003 4\$:
1613 014776 001016
1614 015000 012737 000074 003432
1615 015006 017737 166322 003434
1616 015014 023737 003432 003434
1617 015022 001404
1618 015024
(5) 015024 104455
(6) 015026 000173
(6) 015030 033703
(6) 015032 026220

.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 CSERDF
.WORD 121
.WORD EM74
.WORD ERR2

CKLOOP

TRAP CSCLP1

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 CSERDF
.WORD 122
.WORD EM75
.WORD ERR2

CKLOOP

TRAP CSCLP1

CMP RLTP,#RLV12X
JNE 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 CSERDF
.WORD 123
.WORD EM75A
.WORD ERR2

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 C 6
16 -- UNIQUENESS OF RLBA. PAGE 24-1

SEQ 0067

1619 015034
(3) 015034
(3) 015034 104401

SS: ENDTST

L10032: TRAP CSETST

1621
1622
1623 015036
(2)
1624
1625
1626
1627
1628
1629 015036
(2)
1630 015036
(4) 015036
1631 015036 012737 000200 003432
1632 015044 032777 040000 166252
1633 015052 001403
1634 015054 052737 140000 003432
1635 015062 013777 003432 166234 1S:
1636 015070 012777 177776 166230
1637 015076 023727 003402 000003
1638 015104 001003
1639 015106 012777 000074 166220 2S:
1640 015114 005077 166210
1641 015120 017737 166200 003434
1642 015126 042737 000001 003434
1643 015134 023737 003432 003434
1644 015142 001405
1645 015144 104455
(5) 015144 104455
(6) 015146 000174
(6) 015150 033737
(6) 015152 026220
1646 015154 104406
(3) 015154 104406
1647 015156 012737 177776 003432 3S:
1648 015164 017737 166136 003434
1649 015172 023737 003432 003434
1650 015200 001405
1651 015202 104455
(5) 015202 104455
(6) 015204 000175
(6) 015206 033772
(6) 015210 026220
1652 015212 104406
(3) 015212 104406
1653 015214 023727 003402 000003 4S:
1654 015222 001016
1655 015232 012737 000074 003432
1656 015232 017737 166076 003434
1657 015240 023737 003432 003434
1658 015246 001404
1659 015250 104455
(5) 015250 104455
(6) 015252 000176
(6) 015254 034025
(6) 015256 026220

.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::

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 RLTP,#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 CSERDF
.WORD 124
.WORD EM76
.WORD ERR2

CKLOOP

TRAP CSCLP1

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 CSERDF
.WORD 125
.WORD EM77
.WORD ERR2

CKLOOP

TRAP CSCLP1

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 EM77A,ERR2 : DA MODIFIED BAE.

TRAP CSERDF
.WORD 126
.WORD EM77A
.WORD ERR2

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 ^{E 6} PAGE 25-1
17 -- UNIQUENESS OF RLDA.

SEQ 0069

1660 015260
(3) 015260
(3) 015260 104401

58: ENDTST

L10033: TRAP CSETST

1662
1663
1664 015262
(2)
1665
1666
1667
1668
1669
1670 015262
(2)
1671 015262
(4)
1672 015262 012737 000200 003432
1673 015270 032777 040000 166026
1674 015276 001403
1675 015300 052737 140000 003432
1676 015306 013777 003432 166010 1\$:
1677 015314 012777 177776 166004
1678 015322 012777 177777 166000
1679 015330 023727 003402 000003
1680 015336 001003
1681 015340 012777 000074 165766
1682 015346 005077 165760 2\$:
1683 015352 017737 165746 003434
1684 015360 042737 000001 003434
1685 015366 023737 003432 003434
1686 015374 001405
1687 015376 104455
(5)
1688 015400 000177 TRAP CSERDF
(6) 015402 032333 .WORD 127
(6) 015404 026220 .WORD EM44
1688 015406 104406 .WORD ERR2
(3) 015406 104406 CKLOOP TRAP CSCLP1
1689 015410 012737 177776 003432 3\$:
1690 015416 017737 165704 003434
1691 015424 023737 003432 003434
1692 015432 001405
1693 015434 104455
(5) 015434 104455 TRAP CSERDF
(6) 015436 000200 .WORD 128
(6) 015440 032366 .WORD EM45
(6) 015442 026220 .WORD ERR2
1694 015444 104406 CKLOOP TRAP CSCLP1
(3) 015444 104406
1695 015446 012737 177777 003432 4\$:
1696 015454 017737 165650 003434
1697 015462 023737 003432 003434
1698 015470 001405
1699 015472 104455
(5) 015472 104455 TRAP CSERDF
(6) 015474 000201 .WORD 129
(6) 015476 032421 .WORD EM46
(6) 015500 026220 .WORD ERR2
1700 015502 CKLOOP

```
.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 AND VERIFY THE CONTENTS
:OF THE RLCS, RLBA, RLDA.
: IF RLV12, INCLUDE THE RLBAE ALSO.
STARS
:*****
      BEGIN.TEST
      T18::
1672  MOV    #CRDY,GDDAT    :CONTROLLER READY
1673  BIT    #DERR,@RLCS    :IF DRIVE ERROR IS SET...
1674  BEQ    1$
1675  BIS    #ERR!DERR,GDDAT :...EXPECT IT BACK LATER.
1676  MOV    GDDAT,@RLCS    :LOAD CS
1677  MOV    #-2,@RLBA      :LOAD BA WITH ALL 1'S
1678  MOV    #-1,@RLDA      :LOAD RLDA
1679  CMP    RLTP,#RLV12X
1680  BNE    2$
1681  MOV    #74,@RLBAE     :LOAD RLBAE.
1682  CLR    @RLMP          :WRITE TO RLMP
1683  MOV    @RLCS,BDDAT    :READ RLCS
1684  BIC    #DRDY,BDDAT    :IGNORE DRIVE READY
1685  CMP    GDDAT,BDDAT
1686  BEQ    3$
1687  DFERR  EM44,ERR2      : BR IF CS OK.
                               :MP MODIFIED CS
                               TRAP CSERDF
                               .WORD 127
                               .WORD EM44
                               .WORD ERR2
      CKLOOP
1689  MOV    #-2,GDDAT      : SET EXPECTED BA.
1690  MOV    @RLBA,BDDAT    : READ IT.
1691  CMP    GDDAT,BDDAT
1692  BEQ    4$
1693  DFERR  EM45,ERR2      : BR IF BA OK.
                               :MP MODIFIED BA
                               TRAP CSERDF
                               .WORD 128
                               .WORD EM45
                               .WORD ERR2
      CKLOOP
1695  MOV    #-1,GDDAT      : SET EXPECTED DA.
1696  MOV    @RLDA,BDDAT    : READ IT.
1697  CMP    GDDAT,BDDAT
1698  BEQ    5$
1699  DFERR  EM46,ERR2      : BR IF DA OK.
                               :MP MODIFIED DA
                               TRAP CSERDF
                               .WORD 129
                               .WORD EM46
                               .WORD ERR2
      CKLOOP
```


CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 ⁶ PAGE 26-1
18 -- UNIQUENESS OF RLMP.

SEQ 0071

(3)	015502	104406										
1701	015504	023727	003402	000003	5S:	CMP	RLTYP,#RLV12X			TRAP	CSCLP1	
1702	015512	001016				BNE	6S					
1703	015514	012737	000074	003432		MOV	#74,GDDAT	: SET EXPECTED BAE.				
1704	015522	017737	165606	003434		MOV	@RLBAE,BDDAT	: READ IT.				
1705	015530	023737	003432	003434		CMP	GDDAT,BDDAT					
1706	015536	001404				BEQ	6S	: BR IF BAE OK.				
1707	015540					DFERR	EM47,ERR2	: MP MODIFIED BAE.				
(5)	015540	104455								TRAP	CSERDF	
(6)	015542	000202								.WORD	130	
(6)	015544	032454								.WORD	EM47	
(6)	015546	026220								.WORD	ERR2	
1708	015550				6S:	ENDTST						
(3)	015550											
(3)	015550	104401							L10034:	TRAP	CSETST	

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

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

BEGIN.TEST

T19::

1710
1711
1712 015552
(2)
1713
1714
1715
1716
1717 015552
(2)
1718 015552
(4) 015552
1719 015552 023727 003402 000003
1720 015560 001071
1721 015562 012737 000200 003432
1722 015570 032777 040000 165526
1723 015576 001403
1724 015600 052737 140000 003432
1725 015606 013777 003432 165510 1\$:
1726 015614 005077 165506
1727 015620 005077 165504
1728 015624 012777 000077 165502
1729 015632 052737 000060 003432
1730 015640 017737 165460 003434
1731 015646 042737 000001 003434
1732 015654 023737 003432 003434
1733 015662 001405
1734 015664
(5) 015664 104455
(6) 015666 000203
(6) 015670 032510
(6) 015672 026220
1735 015674
(3) 015674 104406
1736 015676 005037 003432
1737 015702 017737 165420 003434 2\$:
1738 015710 001405
1739 015712
(5) 015712 104455
(6) 015714 000204
(6) 015716 032547
(6) 015720 026220
1740 015722
(3) 015722 104406
1741 015724 017737 165400 003434 3\$:
1742 015732 001404
1743 015734
(5) 015734 104455
(6) 015736 000205
(6) 015740 032573
(6) 015742 026220
1744 015744
(3) 015744
(3) 015744 104401 4\$:

CMP RLTP,#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 CSERDF
.WORD 131
.WORD EM50
.WORD ERR2

CKLOOP

TRAP CSCLP1

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

TRAP CSERDF
.WORD 132
.WORD EM51
.WORD ERR2

CKLOOP

TRAP CSCLP1

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

TRAP CSERDF
.WORD 133
.WORD EM52
.WORD ERR2

ENDTST

L10035:

TRAP CSETST

1746
1747
1748 015746
(2)
1749
1750
1751
1752
1753
1754 015746
(2)
1755 015746
(4) 015746
1756
1757
1758
1759
1760
1761 015746 004537 010714
1762 015752 000000
1763 015754 004537 010210
1764 015760 000000 177001
1765 015764 004537 010640
1766 015770 023727 003402 000000
1767 015776 001047
1768
1769
1770
1771 016000 005037 003356
1772 016004 013737 003350 003432
1773 016012 042737 036000 003432
1774 016020 013737 003364 003434
1775 016026 023737 003432 003434
1776 016034 001405
1777 016036
(5) 016036 104455
(6) 016040 000206
(6) 016042 031405
(6) 016044 026220
1778 016046
(3) 016046 104406
1779 016050 023737 003366 003352 3\$:
1780 016056 001010
1781 016060 023737 003370 003354
1782 016066 001004
1783 016070 023737 003372 003356
1784 016076 001405
1785 016100
(5) 016100 104455
(6) 016102 000207
(6) 016104 031440
(6) 016106 026164
1786 016110
(3) 016110 104406
1787 016112
(3) 016112 104432

```
.SBTTL 20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).  
STARS  
:*****  
: TEST FUNCTION CODE 0.  
: IF RL11 -- NOP. EXPECT CS<13:10> CLEAR, OTHER REGISTERS UNAFFECTED.  
: IF RLVXX -- MAINT. SETUP IN NORMAL FASHION, BUT ONLY CHECK FOR  
: FUNCTION COMPLETE (NO ERRORS AND DA = INITIAL DA+6).  
: WE'LL CHECK THE REST OF THE MAINT FUNCTION LATER.  
$STARS  
:*****  
: BEGIN.TEST  
T20::  
: INITIAL SETUP ASSUMES THAT WE HAVE AN RLV.  
: IF NOT -- NOT TO WORRY -- IT SHOULD JUST NO-OP.  
: ALL WE CARE ABOUT IS THAT WE GET NO ERRORS (OR HUNG RL).  
1$: JSR R5,CALCRC ; MAKE A CRC EVEN THO WE WON'T CHECK IT.  
: .WORD 0 ; THIS WILL BE THE INITIAL DA WORD.  
: JSR R5,LDFUN ; LOAD UP THE FUNCTION.  
: 0, -511. ; FUNCTION 0, WC -511.  
: JSR R5,WTCRDY ; WAIT FOR CONTROLLER.  
: CMP RL1YP,#RL11 ; NOW ARE WE RL11 ??  
: BNE 6$ ; CHECK 'MAINT' FINAL STATE IF NOT.  
: RL11 -- CHECK FOR ERRORS AFTER THE NO-OP.  
2$: CLR B.MP ; MPR WILL BE 0.  
: MOV B,CS,GDDAT ; EXPECT 'NO-OP' SHOULD HAVE  
: BIC #036000,GDDAT ; ...CLEARED ANY ERROR BITS <13:10>...  
: MOV E,CS,BDDAT ; ... THAT MIGHT HAVE BEEN ON.  
: CMP GDDAT,BDDAT ; IS THAT WHAT WE HAVE ??  
: BEQ 3$ ; YES.  
: DFERR EM14,ERR2 ; CS WRONG AFTER NOP.  
TRAP CSERDF  
.WORD 134  
.WORD EM14  
.WORD ERR2  
CKLOOP  
TRAP CSCLP1  
3$: CMP E,BA,B.BA ; BA UNALTERED ??  
: BNE 4$  
: CMP E,DA,B.DA ; DA UNALTERED ??  
: BNE 4$  
: CMP E,MP,B.MP ; MP UNALTERED ??  
: BEQ 5$ ; EXIT IF ALL THREE WERE OK.  
: DFERR EM14A,ERRO ; REGISTERS ALTERED BY NOP.  
TRAP CSERDF  
.WORD 135  
.WORD EM14A  
.WORD ERRO  
CKLOOP  
TRAP CSCLP1  
5$: EXIT TST ; RL NOP TEST DONE.  
TRAP CSEXIT
```


CVRLBAO -- RLV12 DISKLESS.
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 J 6 PAGE 28-1
20 -- FUNCTION CODE 0, NOP (RL11), OR MAINT (RLV11/12).

SEQ 0074

```
(3) 016114 000042 .WORD L10036-.
1788
1789
1790
1791 016116 013737 003370 003434
1792 016124 023727 003434 000006
1793 016132 001004
1794 016134 032737 002000 003364
1795 016142 001405
1796 016144
(5) 016144 104455
(6) 016146 000210
(6) 016150 031502
(6) 016152 026254
1797 016154
(3) 016154 104406
1798 016156
(3) 016156
(3) 016156 104401

: RLV11 OR 12 -- CHECK SEQUENCER STATE AFTER INTERNAL DIAGNOSTIC.
6S: MOV E,DA,BDDAT ; GET FINAL SEQUENCER STATE.
CMP BDDAT,#6 ; FINAL STATE RIGHT ??
BNE 7S ; NO ERROR.
BIT #OPI,E.CS ; YES, OPI SHOULD BE CLEAR.
BEQ 8S ; IF IT IS, WE'RE FAT !!
7S: DFERR EM14B,ERR3 ; MAINT FAILURE, DA = SEQUENCER STATE.
TRAP CSERDF
.WORD 136
.WORD EM14B
.WORD ERR3
CKLOOP
TRAP CSCLP1
8S: ENDTST
L10036: TRAP CSETST
```

1800
1801
1802 016160
(2)
1803
1804
1805
1806
1807
1808
1809 016160
(2)
1810 016160
(4) 016160
1811 016160 005037 010634
1812 016164 012700 000000
(3) 016164 104441
(3) 016170 104441
1813 016172 004537 010714
1814 016176 000000
1815 016200 004537 010210
1816 016204 000100 177001
1817 016210 004537 010640
1818 016214 005737 010634
1819 016220 001006
1820 016222 005037 003424
1821 016226 104455
(5) 016226 000211
(6) 016230 031544
(6) 016232 026370
(6) 016234
1822 016236 012700 000340
(3) 016236 104441
(3) 016242
1823 016244
(3) 016244
(3) 016244 104401

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

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 T21::
SETPRI #PRI00 ;SET PSW TO 0
JSR R5,CALCRC ; AGAIN, CRC WON'T BE CHECKED.
.WORD 0
JSR R5,LDLDFUN ; LOAD UP AND EXECUTE...
INTEN!0,-511 ; NOP/MAINT AND INTERRUPT.
JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY.
TST INTFLG ;DID INTERRUPT OCCUR ??
BNE IS ; BR IF SO.
CLR TMPO ; CPU LEVEL FOR ERROR MSG.
DFERR EM15,ERR7 ; INTERRUPT NOT REC'D.

MOV #PRI00,R0
TRAP CSSPRI

TRAP CSERDF
.WORD 137
.WORD EM15
.WORD ERR7

IS: SETPRI #PRI07

MOV #PRI07,R0
TRAP CSSPRI

ENDTST

L10037: TRAP CSETST

1825
1826
1827 016246
(2)
1828
1829
1830
1831
1832
1833
1834 016246
(2)
1835 016246
(4) 016246
1836 016246 012737 000340 003434
1837 016254 012737 000007 003424
1838 016262 013737 003340 003432
1839 016270
(3) 016270 104404
1840 016272
(3) 016272 013700 003434
(3) 016276 104441
1841 016300 005037 010634
1842 016304 004537 010714
1843 016310 000000
1844 016312 004537 010210
1845 016316 000100 177001
1846 016322 004537 010640
1847 016326 023737 003434 003432
1848 016334 002004
1849 016336 005737 010634
1850 016342 001011
1851 016344 000403
1852 016346 005737 010634
1853 016352 001405
1854 016354
(5) 016354 104455
(6) 016356 000212
(6) 016360 031606
(6) 016362 026370
1855 016364
(3) 016364 104406
1856 016366 162737 000040 003434 4\$:
1857 016374 005337 003424
1858 016400 100334
1859 016402
(3) 016402
(3) 016402 104405
1860 016404
(3) 016404 012700 000340
(3) 016410 104441
1861 016412
(3) 016412
(3) 016412 104401

.SBTTL 22 -- TEST INTERRUPT PRIORITY BR LEVEL.

STARS

:TEST THAT PRIORITY GIVEN IS ACTUAL PRIORITY OF CONTROLLER.
:WE ALREADY KNOW THAT THE CONTROLLER CAN INTERRUPT. WE'LL START
:AT CPU LEVEL 7 AND WORK DOWN 'TIL THE INTERRUPT COMES IN.
:AS BEFORE, WE'RE USING THE NOP/MAINT FUNCTION.
:NOTE THAT RLV11 IS FIXED AT BR4, RL11 AND RLV12 ARE PROGRAMABLE
:AND DEFAULT TO BR5. LOC 'BPRIOR' IS PROPERLY SET AT INIT TIME.
STARS

BEGIN.TEST

T22::

MOV #PRI07,BDDAT
MOV #7,TMP0
MOV BPRIOR,GDDAT
BGNSEG

:SET UP INITIAL OF 7
: A COPY FOR ERROR MSG.
:GET GIVEN PRIORITY
: **** START SEGMENT ****

1\$: SETPRI BDDAT

: SET CURRENT PRIORITY...

TRAP CSBSEG
MOV BDDAT,RO
TRAP CSSPRI

CLR INTFLG
JSR R5,CALCRC
.WORD 0
JSR R5,LDLUN

:...AND CLEAR INTERRUPT FLAG.
: JUST LIKE BEFORE.

INTEN!0,-511
JSR R5,WTCRDY
CMP BDDAT,GDDAT

: LOAD AND EXECUTE...
:...NOP/MAINT AND INTERRUPT.
:WAIT FOR CONTROLLER READY.

BGE 2\$
TST INTFLG
BNE 4\$
BR 3\$

:SHOULD IT INTERRUPT
: BR IF NOT.
: DID IT INTERRUPT ??
: PROCEED IF SO.
: NO, ERROR.
: DID IT INTERRUPT ??
: PROCEED IF NOT.

2\$: TST INTFLG

3\$: DFERR EM16,ERR7

: INTERRUPT PRIORITY FAILURE.

TRAP CSERDF
.WORD 138
.WORD EM16
.WORD ERR7

CKLOOP

SUB #40,BDDAT
DEC TMP0
BPL 1\$
ENDSEG

:LOWER CPU PRIORITY...
:...TO NEXT LEVEL.

4\$:

10000\$:

SETPRI #PRI07

TRAP CSESEG

ENDTST

MOV #PRI07,RO
TRAP CSSPRI

L10040:

TRAP CSETST

1863
1864
1865 016414
(2)
1866
1867
1868
1869
1870 016414
(2)
1871 016414
(4) 016414
1872
1873
1874
1875
1876 016414 004537 010714
1877 016420 000000
1878 016422 012737 177003 016452
1879 016430 112737 000074 032251
1880 016436 112737 000060 032255
1881 016444 004537 010210
1882 016450 000000
1883 016452 177003
1884 016454 004537 010640
1885 016460 012700 112000
1886 016464 043700 003364
1887 016470 001410
1888 016472 004537 007510
1889 016476 000240
1890 016500
(5) 016500 104455
(6) 016502 000213
(6) 016504 032151
(6) 016506 026420
1891 016510
(3) 016510 104406
1892 016512 023727 016452 177000 4\$:
1893 016520 001412
1894 016522 012737 177000 016452
1895 016530 112737 000076 032251
1896 016536 112737 000061 032255
1897 016544 000737
1898 016546
(3) 016546
(3) 016546 104401

.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\$
 JSR R5,GETERR ; EXIT IF WE'RE OK.
 NOP ; OTHERWISE, GET THEM ALL...
 DFERR EM27,ERR10 ;... NONE IS EQUALLY BAD NEWS.
 ; STATUS INCORRECT, WC <> 511.

TRAP CSERDF
.WORD 139
.WORD EM27
.WORD ERR10

CKLOOP

TRAP CSCLP1

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 CSETST

CVRLBA0 -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 N 6 PAGE 33
24 -- RLV11/12 MAINTENANCE, FORCED OPI INTERRUPT.

SEQ 0078

1901
1902
1903 016550
(2)
1904
1905
1906
1907 016550
(2)
1908 016550
(4) 016550
1909 016550 004537 010714
1910 016554 000000
1911 016556 012700 000000
(3) 016556 104441
(3) 016562 104441
1912 016564 005037 010634
1913 016570 004537 010210
1914 016574 000100 177003
1915 016600 004537 010640
1916 016604 005737 010634
1917 016610 001005
1918 016612 104455
(5) 016612 000214
(6) 016614 031641
(6) 016616 026164
(6) 016620 104406
1919 016622 104406
(3) 016622 104406
1920 016624 012700 000340
(3) 016624 104441
(3) 016630 104441
1921 016632
(3) 016632
(3) 016632 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

1\$: JSR R5,CALCRC ;CALCULATE CRC FOR... T24::
.WORD 0 ;... INITIAL DA OF ZERO.
SETPRI #PRI00 ;SET PRIORITY TO ZERO
MOV #PRI00,R0
TRAP CSSPRI
CLR INTFLG ;CLEAR INT. FLAG
JSR R5,LDGUN
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 CSERDF
.WORD 140
.WORD EM17
.WORD ERRO
CKLOOP
TRAP CSCLP1
4\$: SETPRI #PRI07
MOV #PRI07,R0
TRAP CSSPRI
ENDTST
L10042:
TRAP CSETST

1923
1924
1925 016634
(2)
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935 016634
(2)
1936 016634
(4) 016634
1937 016634 005737 003400
1938 016640 100447
1939 016642 004537 010714
1940 016646 000000
1941 016650 012704 000002
1942 016654 004537 010210
1943 016660 000000 177003
1944 016664 004537 010640
1945 016670 012737 001440 003434
1946 016676 166637 177774 003434
1947 016704 032777 002000 164412
1948 016712 001005
1949 016714
(5) 016714 104455
(6) 016716 000215
(6) 016720 032260
(6) 016722 026164
1950 016724 000415
1951 016726 023737 003434 003450 3\$:
1952 016734 003004
1953 016736 023737 003434 003446
1954 016744 002005
1955 016746 077436
1956 016750
(5) 016750 104455
(6) 016752 000216
(6) 016754 032306
(6) 016756 026450
1957 016760
(3) 016760
(3) 016760 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

1\$: TST CPUTYP ; LSI CPU ??? T25::
BMI 5\$; IF NOT, DON'T EVEN TRY !!!
JSR R5,CALCRC ; MAKE CRC FOR ZERO.
.WORD 0
2\$: MOV #2,R4 ;PERFORM MAINT. FUNCTION
JSR R5, LDFUN ; MAINT WITH INVALID WC.
MAINT, -509. ; WAIT FOR DONE.
JSR R5, WTCRDY ; CALCULATE OPI TIME...
MOV #800, BDDAT ; IN BDDAT.
SUB -4(SP), BDDAT ; OPI ERROR ??
BIT #OPI, @RLCS ; YES, CHECK TIMING.
BNE 3\$; OPI FLAG NOT RECEIVED.
DFERR EM31, ERRO

TRAP CSERDF
.WORD 141
.WORD EM31
.WORD ERRO

3\$: BR 5\$
CMP BDDAT, OPIMX
BGT 4\$
4\$: CMP BDDAT, OPIMN ; OK IF TIME WITHIN LIMITS.
BGE 5\$; TRY ONCE MORE.
SOB R4, 2\$; OPI TIMING INCORRECT
DFERR EM32, ERR11

TRAP CSERDF
.WORD 142
.WORD EM32
.WORD ERR11

5\$: ENDTST

L10043:

TRAP CSETST

1959
1960
1961 016762
(2)
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973 016762
(2)
1974 016762
(4) 016762
1975
1976
1977
1978 016762
(3) 016762
(3) 016762 104402
1979 016764 012704 004200
1980 016770 016437 000072 017002
1981 016776 004537 010356
1982 017002 000000
1983 017004 012437 017014
1984 017010 004537 010714
1985 017014 000000
1986 017016 004737 017132
1987 017022 005714
1988 017024 001361
1989 017026
(3) 017026
(3) 017026 104403
1990
1991
1992
1993 017030
(3) 017030
(3) 017030 104402
1994 017032 012704 004200
1995 017036 016437 000072 017050
1996 017044 004537 010362
1997 017050 000000
1998 017052 012437 017062
1999 017056 004537 010714
2000 017062 000000
2001 017064 004737 017132
2002 017070 005714
2003 017072 001361
2004 017074
(3) 017074

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

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.TEST

T26::

:SUBTEST 1 -- 28 DIFFERENT DATA PATTERNS.

DMA1: BGNSUB

T26.1:

TRAP CSBSUB

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 CSSESUB

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

DMA2: BGNSUB

T26.2:

TRAP CSBSUB

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) 017074 104403                                TRAP  CSESUB
2005
2006      : SUBTEST 3 -- ONE RANDOM DATA PATTERN.
2007
2008 017076      DMA3:  BGNSUB                                T26.3:
(3) 017076      (3) 017076      104402                                TRAP  CSBSUB
2009 017100      004537      010452      017116      JSR    R5,SETRAN      :FILL BUFFER WITH RANDOM DATA.
2010 017104      013737      003456      MOV    TEMLO,1$      : USE TEMLO FOR THE CRC WORD.
2011 017112      004537      010714      JSR    R5,CALCRC      : CALCULATE CRC'S.
2012 017116      000000      1$:      .WORD 0
2013 017120      004737      017132      JSR    PC,EXECUT      : XCT
2014 017124
(3) 017124
(3) 017124      104403                                L10047:
2015 017126      EXIT    TST                                TRAP  CSESUB
(3) 017126      104432                                TRAP  CSEXIT
(3) 017130      000530                                .WORD L10044-.
2016
2017      : THIS SUBROUTINE EXECUTES THE FUNCTION AND VERIFIES RESULTS.
2018
2019      EXECUT:  BGNSEG      : *** SET LOOP SEGMENT ***
(3) 017132      104404                                TRAP  CSBSEG
2020 017134      004537      010210      JSR    R5,LDFUN      :PERFORM MAINT. FUNCTION
2021 017140      000000      177001      MAINT, -511.
2022 017144      004537      010640      JSR    R5,WTCRDY      :WAIT FOR READY
2023 017150      004537      007510      JSR    R5,GETERR      :CHECK CONTROLLER FOR ERRORS
2024 017154      000405      BR     11$          : BR IF NONE.
2025 017156      DFERR   EM98,ERR6      :UNEXPECTED CONTROLLER ERRORS.
(5) 017156      104455                                TRAP  CSERDF
(6) 017160      000217                                .WORD 143
(6) 017162      034347                                .WORD EM98
(6) 017164      026340                                .WORD ERR6
2026 017166      CKLOOP
(3) 017166      104406                                TRAP  CSCLP1
2027 017170      032737      022000      003364      11$:  BIT    #OPI!NXM,E.CS      : OPI OR NXM ??
2028 017176      001402      BEQ    12$          : PROCEED IF NOT.
2029 017200      EXIT    SEG      : YES, DON'T BOTHER WITH ANY MORE.
(3) 017200      104432                                TRAP  CSEXIT
(3) 017202      000452                                .WORD 10000S-.
2030 017204      012737      006362      003432      12$:  MOV    #BUFEND-2,GDDAT
2031 017212      013737      003366      003434      MOV    E.BA,BDDAT
2032 017220      023737      003432      003434      CMP    GDDAT,BDDAT      :FINAL BA CORRECT ??
2033 017226      001405      BEQ    1$
2034 017230      DFERR   EM10,ERR4      :FINAL BA INCORRECT.
(5) 017230      104455                                TRAP  CSERDF
(6) 017232      000220                                .WORD 144
(6) 017234      031303                                .WORD EM10
(6) 017236      026304                                .WORD ERR4
2035 017240      CKLOOP
(3) 017240      104406                                TRAP  CSCLP1
2036 017242      013737      003354      003432      1$:  MOV    B.DA,GDDAT      :GET BEFORE DA REGISTER
2037 017250      013737      003370      003434      MOV    E.DA,BDDAT
2038 017256      005037      003424      CLR    TMPO
2039 017262      113737      003354      003424      MOVB   B.DA,TMPO
2040 017270      062737      000006      003424      ADD    #6,TMPO          :+6 TO DA LOW BYTE

```


2041	017276	113737	003424	003432		MOVB	TMPO,GDDAT	:STORE LOW BYTE OF DA		
2042	017304	023737	003432	003434		CMP	GDDAT,BDDAT	: FINAL DA CORRECT ??		
2043	017312	001405				BEQ	2\$			
2044	017314					DFERR	EM12,ERR4	: FINAL DA INCORRECT.		
(5)	017314	104455							TRAP	C\$ERDF
(6)	017316	000221							.WORD	145
(6)	017320	031324							.WORD	EM12
(6)	017322	026304							.WORD	ERR4
2045	017324					CKLOOP				
(3)	017324	104406							TRAP	C\$CLP1
2046	017326	013737	003442	003432	2\$:	MOV	GDCRC3,GDDAT	:GET EXPECTED CRC OF DA+3 VALUE		
2047	017334	013737	003372	003434		MOV	E.MP1,BDDAT	:GET CONTROLLER CRC OF DA+3		
2048	017342	023737	003432	003434		CMP	GDDAT,BDDAT			
2049	017350	001405				BEQ	3\$			
2050	017352					DFERR	EM20,ERR4	:CRC DA+3 INCORRECT.		
(5)	017352	104455							TRAP	C\$ERDF
(6)	017354	000222							.WORD	146
(6)	017356	031706							.WORD	EM20
(6)	017360	026304							.WORD	ERR4
2051	017362					CKLOOP				
(3)	017362	104406							TRAP	C\$CLP1
2052	017364	013737	003444	003432	3\$:	MOV	GDCRC4,GDDAT			
2053	017372	013737	003374	003434		MOV	E.MP1,BDDAT			
2054	017400	023737	003432	003434		CMP	GDDAT,BDDAT	: DITTO CRC OF CRC OF DA+4.		
2055	017406	001405				BEQ	4\$			
2056	017410					DFERR	EM21,ERR4	:CRC OF CRC OF DA+4 INCORRECT.		
(5)	017410	104455							TRAP	C\$ERDF
(6)	017412	000223							.WORD	147
(6)	017414	031757							.WORD	EM21
(6)	017416	026304							.WORD	ERR4
2057	017420					CKLOOP				
(3)	017420	104406							TRAP	C\$CLP1
2058	017422	005037	003430		4\$:	CLR	TMP2	:CLEAR BAD WORD COUNTER		
2059	017424	012703	004364			MOV	#BUF1,R3	:GOOD DATA STORED IN BUF1		
2060	017432	012702	005364			MOV	#BUF2,R2	:DATA BUFFER WRITTEN INTO BY MAINT.		
2061	017436	005001				CLR	R1	: WORD COUNT.		
2062	017440	011337	003432		5\$:	MOV	(R3),GDDAT	:EXPECTED DATA		
2063	017444	011237	003434			MOV	(R2),BDDAT	:GET DATA FROM BUFFER		
2064	017450	023737	003432	003434		CMP	GDDAT,BDDAT			
2065	017456	001436				BEQ	7\$:BR IF DATA IS RIGHT.		
2066	017460	010237	003424			MOV	R2,TMP0	: GET ERROR ADDRESS...		
2067	017464	010137	003426			MOV	R1,TMP1	:...AND WORD NUMBER.		
2068	017470	005737	003430			TST	TMP2	: 1ST ERROR ENCOUNTERED ??		
2069	017474	001004				BNE	6\$:NO, SKIP THE ERROR HEADER.		
2070	017476					DFERR	EM22,ERR0	:ERROR MESSAGE ON 1ST ERROR...		
(5)	017476	104455							TRAP	C\$ERDF
(6)	017500	000224							.WORD	148
(6)	017502	032037							.WORD	EM22
(6)	017504	026164							.WORD	ERR0
2071	017506	005237	003430		6\$:	INC	TMP2	:...FOLLOWED BY FAILING DATA.		
2072	017512					PRINTX	#FRMT14,TMP1,TMP0,GDDAT,BDDAT			
(11)	017512	013746	003434						MOV	BDDAT,-(SP)
(10)	017516	013746	003432						MOV	GDDAT,-(SP)
(9)	017522	013746	003424						MOV	TMP0,-(SP)
(8)	017526	013746	003426						MOV	TMP1,-(SP)
(7)	017532	012746	030031						MOV	#FRMT14,-(SP)

Address	Offset	Value	Label	Instruction	Comment	Trap	Trap Action
(6)	017536	012746	000005			MOV	#5,-(SP)
(3)	017542	010600				MOV	SP,R0
(4)	017544	104415				TRAP	CSPNTX
(4)	017546	062706	000014			ADD	#14,SP
2073	017552			CKLOOP			
(3)	017552	104406				TRAP	CSCLP1
2074	017554	005722		7\$: TST (R2)+	: INCREMENT BUFFER POINTERS.		
2075	017556	005723		TST (R3)+			
2076	017560	005201		INC R1	: AND THE WORD COUNT.		
2077	017562	020127	000377	CMP R1,#255.	: DONE ALL WORDS ??		
2078	017566	002724		BLT 5\$: NOT YET, CONTINUE.		
2079	017570	005737	003430	TST TMP2	: YES, ANY ERRORS LOGGED ??		
2080	017574	001412		BEQ 8\$: NO.		
2081	017576			PRINTB #FRMT15,TMP2	: YES, PRINT SUMMARY.		
(8)	017576	013746	003430			MOV	TMP2,-(SP)
(7)	017602	012746	030120			MOV	#FRMT15,-(SP)
(6)	017606	012746	000002			MOV	#2,-(SP)
(3)	017612	010600				MOV	SP,R0
(4)	017614	104414				TRAP	CSPNTB
(4)	017616	062706	000006			ADD	#6,SP
2082	017622	012737	123456	003432 8\$: MOV #123456,GDDAT	: EXPECTED DATA IN LAST WORD+1		
2083	017630	011237	003434	MOV (R2),BDDAT	: GET LAST WORD+1 FROM BUF2		
2084	017634	023737	003432	CMP GDDAT,BDDAT			
2085	017642	001404		BEQ 9\$			
2086	017644			DFERR EM23,ERR4	: LAST+1 INCORRECT.		
(5)	017644	104455				TRAP	CSEPDF
(6)	017646	000225				.WORD	149
(6)	017650	032103				.WORD	EM23
(6)	017652	026304				.WORD	ERR4
2087	017654			9\$: ENDSEG			
(3)	017654					10000\$:	
(3)	017654	104405				TRAP	CSESEG
2088	017656	000207		RTS PC	: RETURN.		
2089							
2090	017660			10\$: ENDTST			
(3)	017660					L10044:	
(3)	017660	104401				TRAP	CSETST

```
2092 .SBTTL 27 -- RLV11/12 MAINTENANCE, FIFO ADDRESSING.
2093
2094 017662 STARS
(2) :*****
2095 :TEST THAT FIFO OPERATES CORRECTLY. STORE ADDRESS PATTERN
2096 :IN BUF1 (0-255) THAT CONTAINS A UNIQUE PATTERN IN EACH LOCATION.
2097 :PERFORM MAINTENANCE FUNCTION AND TEST BUF2 FOR PROPER FIFO
2098 :ADDRESSING. NOTE THAT CRC'S ARE NOT CHECKED IN THIS TEST.
2099 :REPEAT A SECOND TIME USING A COMPLIMENT ADDRESS PATTERN.
2100 :ALSO USE AND TEST FOR INTERRUPT ON MAINT DONE.
2101 017662 STARS
(2) :*****
2102 017662 :
(4) 017662 BEGIN.TEST
2103
2104 :
2105 : SUBTEST 1 -- FIFO ADDRESS PATTERN.
2106 017662 FIFO1: BGNSUB
(3) 017662
(3) 017662 104402 T27.1: TRAP CSBSUB
2107 017664 005001 CLR R1
2108 017666 012702 000400 MOV #256.,R2
2109 017672 012703 004364 MOV #BUF1,R3
2110 017676 010123 1$: MOV R1,(R3)+ ;SETUP TO STORE PATTERN IN BUF1
2111 017700 005201 INC R1 ;INC. PATTERN
2112 017702 005302 DEC R2
2113 017704 001374 BNE 1$
2114 017706 004737 017754 JSR PC,XFIFO ; EXECUTE AND CHECK FIFO DATA.
2115 017712 ENDSUB
(3) 017712
(3) 017712 104403 L10051: TRAP CSESUB
2116
2117 :
2118 : SUBTEST 2 -- FIFO COMPLIMENT ADDRESS PATTERN.
2119 017714 FIFO2: BGNSUB
(3) 017714
(3) 017714 104402 T27.2: TRAP CSBSUB
2120 017716 012701 177777 MOV #-1,R1
2121 017722 012702 000400 MOV #256.,R2
2122 017726 012703 004364 MOV #BUF1,R3
2123 017732 010123 1$: MOV R1,(R3)+ ; STORE COMPLIMENT ADDRESS PATTERN.
2124 017734 005301 DEC R1
2125 017736 005302 DEC R2
2126 017740 001374 BNE 1$
2127 017742 004737 017754 JSR PC,XFIFO ; EXECUTE AND CHECK FIFO DATA.
2128 017746 ENDSUB
(3) 017746
(3) 017746 104403 L10052: TRAP CSESUB
2129 017750 EXIT TST ; ALL DONE.
(3) 017750 104432 TRAP CSEXIT
(3) 017752 000332 .WORD L10050-.
2130
2131 :
2132 : THIS SUBROUTINE EXECUTES THE MAINT FUNCTION AND TESTS RESULTS.
2133 017754 XFIFO: BGNSEG ;*** LOOP SEGMENT ***
(3) 017754 104404 TRAP CSBSEG
```

```
2134 017756 012702 000400      MOV    #256,R2      :SETUP TO CLEAR BUF2
2135 017762 012703 005364      MOV    #BUF2,R3
2136 017766 005023      1$:   CLR    (R3)+
2137 017770 005302      DEC    R2
2138 017772 001372      BNE    1$
2139 017774 005037 010634      CLR    INTFLG      :CLEAR INT. FLAG
2140 020000      SETPRI #PRI00
2141 (3) 020000 012700 000000      MOV    #PRI00,R0
2142 (3) 020004 104441      TRAP  CSSPRI
2143 020006 004537 010210      JSR    R5,LDFUN    :LOAD FUNCTION
2144 020012 000100 177001      MAINT!INTEN,-511.
2145 020016 004537 010640      JSR    R5,WTCRDY  :WAIT FOR READY
2146 020022 004537 007510      JSR    R5,GETERR  :CHECK CONTROLLER FOR ERRORS
2147 020026 000405      BR     2$          :BR IF NONE.
2148 (5) 020030 104455      DFERR  EM98,ERR6  :UNEXPECTED CONTROLLER ERRORS.
2149 (6) 020032 000226      TRAP  CSERDF
2150 (6) 020034 034347      .WORD 150
2151 (6) 020036 026340      .WORD EM98
2152 (3) 020040      .WORD  ERR6
2153 020042 104406      CKLOOP
2154 020044 032737 022000 003364 2$:   BIT    #OPI!NXM,E.CS : OPI OR NXM ??
2155 020050 001110      BNE    8$          : QUIT IF SO.
2156 020052 005737 010634      TST    INTFLG     :CHECK FOR INTERRUPT
2157 020056 001005      BNE    4$
2158 (5) 020060 104455      DFERR  EM15,ERRO  : DIDN'T INTERRUPT.
2159 (6) 020062 000227      TRAP  CSERDF
2160 (6) 020064 031544      .WORD 151
2161 (6) 020066 026164      .WORD EM15
2162 (3) 020070      .WORD  ERRO
2163 020072 104406      CKLOOP
2164 020074 005037 003430      4$:   CLR    TMP2
2165 020076 012703 004364      MOV    #BUF1,R3   :CLEAR BAD WORD COUNTER
2166 020102 012702 005364      MOV    #BUF2,R2   :GOOD DATA STORED IN BUF1
2167 020106 005001      CLR    R1         :DATA BUFFER WRITTEN INTO BY MAINT.
2168 020110 011337 003432      5$:   MOV    (R3),GDDAT : WORD COUNT.
2169 020114 011237 003434      MOV    (R2),BDDAT : EXPECTED DATA
2170 020120 023737 003432 003434      CMP    GDDAT,BDDAT :GET DATA FROM BUFFER
2171 020126 001436      BEQ    7$
2172 020130 010237 003424      MOV    R2,TMP0    :BR IF DATA IS RIGHT.
2173 020134 010137 003426      MOV    R1,TMP1    : GET ERROR ADDRESS...
2174 020140 005737 003430      TST    TMP2       :...AND WORD NUMBER.
2175 020144 001004      BNE    6$         : 1ST ERROR ENCOUNTERED ??
2176 (5) 020146 104455      DFERR  EM22,ERRO  :NO, SKIP THE ERROR HEADER.
2177 (6) 020150 000230      .WORD 152
2178 (6) 020152 032037      .WORD EM22
2179 (6) 020154 026164      .WORD  ERRO
2180 020156 005237 003430      6$:   INC    TMP2
2181 (11) 020162 013746 003434      PRINTX #FRMT14,TMP1,TMP0,GDDAT,BDDAT :...FOLLOWED BY FAILING DATA.
2182 (10) 020166 013746 003432      MOV    BDDAT,-(SP)
2183 (9) 020172 013746 003424      MOV    GDDAT,-(SP)
2184 (8) 020176 013746 003426      MOV    TMP0,-(SP)
2185 (7) 020202 012746 030031      MOV    TMP1,-(SP)
2186 (7) 020202 012746 030031      MOV    #FRMT14,-(SP)
```


2184
2185
2186 020306
(2)
2187
2188
2189
2190
2191
2192
2193
2194
2195 020306
(2)
2196 020306
(4)
2197 020306
2198 020314
2199 020316
2200 020324
2201 020326
(3)
2202 020330
2203 020332
2204 020340
2205 020342
2206 020346
2207 020352
2208 020354
2209 020362
2210 020366
2211 020374
2212 020376
2213 020402
2214 020406
2215 020412
2216 020416
2217 020422
2218 020426
2219 020432
2220 020434
2221 020440
2222 020442
2223 020450
2224 020456
(5)
(6)
(6)
(6)
2225
2226 020466
2227 020474
2228 020500
(3)
(3)

020306
020306
023727 003402 000001
003006
023727 011232 000174
002406
104432
000150
112737 000054 034112 1\$:
000402
105037 034112 2\$:
004537 010714
000000
012737 160000 010352
012700 000003
023727 011240 000026
001002
012700 000077
010037 010354 3\$:
004537 010210
000000 177001
004537 010640 BB7:
012700 120000
043700 003364
001415
004537 007510
000240
013737 010354 003426
013737 010352 003430
104455
000231
034061
026500
012737 004364 010352 4\$:
005037 010354
104401

```
.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 CPUNCHED !!!!
* FOR RLV11:
* RLV11 DOESN'T ASSERT BBS7, SO WE'LL THIS TEST
* IF PHYSICAL MEMORY SIZE IS 124K OR GREATER.
STARS
*****
BEGIN.TEST
T28::
CMP RLTP,#RLV11
BGT 1$ ; EXECUTE ALWAYS ON RLV12...
CMP MSIZE,#124. ;...AND ON RLV11 IF < 124KW.
BLT 2$
EXIT TST
TRAP .WORD CSEXIT
L10053-.
1$: MOV#',,EMBOX ; ADJUST ERROR TEXT FOR RLV12...
SKP2
CLR#EMBOX ;...OR FOR RLV11.
JSR R5,CALCRC ; DO CRC FOR INITIAL...
;...DA OF 0.
MOV#160000,BA16 ; POINT TO LOC 0 IN LAST 4K.
; SET AN 18 BIT...
#3,RO
CMP ABUSW,#22.
BNE 3$ ;...OR 22 BIT EXTENSION.
MOV#77,RO
MOV RO,BA22 ; XCT MAINTENANCE...
;...WITH VALID WORD COUNT.
JSR R5,LDFUN
MAINT,-511.
BB7: JSR R5,WTCRDY ; EXPECT AT LEAST THESE ERRORS.
MOV#ERR!NXM,RO
BIC E.CS,RO ; EXIT IF WE'RE OK.
BEQ 4$
JSR R5,GETERR ; OTHERWISE, GET WHAT'S THERE.
NOP
MOV BA22,TMP1 ; AND GET PHYSICAL ADDRESS.
MOV BA16,TMP2
DFERR EM80,ERR12 ; BBS7 AND/OR NXM FAILURE.
TRAP .WORD CSERDF
.WORD 153
.WORD EM80
.WORD ERR12
4$: MOV#BUF1,BA16 ; RESET BA16 AND 22.
CLR BA22
ENDTST
L10053: TRAP CSETST
```

```
2230 .SBTTL 29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST.
2231
2232 020502 STARS
(2) :*****
2233 :* BYPASS THIS TEST IF NO KT SUPPORT.
2234 :* OTHERWISE, MEMORY HAS ALREADY BEEN SIZED.
2235 :*PAR'S 0 TO 6 ARE MAPPED TO THE 1ST 28K, AND PAR7 TO THE I/O PAGE.
2236 :*
2237 :*NOW ACCESS EXTENDED MEMORY IN 4K INCREMENTS. STORE A RANDOM DATA
2238 :*PATTERN IN BUF1 (RELOCATED). EXECUTE MAINTENANCE MODE, AND CHECK
2239 :*BUF2 (RELOCATED) FOR CORRECT DATA TRANSFER. ALSO CHECK THAT THE FINAL
2240 :*BUS ADDRESS INCREMENTED CORRECTLY (OVERFLOWING INTO THE BAE).
2241 :*WHEN THE END OF PHYSICAL MEMORY IS REACHED, A BUS-ERROR WILL OCCUR.
2242 :*DURING THE BUFFER SET-UP PHASE AND WE'LL LOOK FOR A NXM ERROR AT
2243 :*THE SAME PHYSICAL ADDRESS. CONTINUE UNTIL WE REACH THE I/O PAGE
2244 :*AT 760000 (18 BIT), OR 17760000 (22 BIT).
2245 020502 STARS
(2) :*****
2246 172350 PAR4= KIPAR0+10 ; USE KIPAR4 FOR ACCESSING.
2247 020502 BEGIN.TEST
(4) 020502
2248 020502 005037 010626 CLR TRPFLG ; CLEAR TRAP FLAG. T29::
2249 020506 005737 177572 TST MMRO ; KT AVAILABLE ??
2250 020512 000240 240
2251 020514 005737 010626 TST TRPFLG
2252 020520 001402 BEQ EXMTST ; PROCEED IF SO.
2253 020522 1$: EXIT TST ;...EXIT OTHERWISE.
(3) 020524 104432
(3) 020524 000550 TRAP WORD C$EXIT
2254 020526 012737 007400 021252 EXMTST: MOV #7400,IOPAG ; ASSUME 18 BITS, SET LIMIT AT 124K...
2255 020534 023727 011240 000026 CMP .ABUSW,#22. ; 22 BITS AVAILABLE ??
2256 020542 001012 BNE 1$ ; BR IF NOT.
2257 020544 023727 003402 000003 CMP RLTP,#RLV12X ; CAN WE HANDLE THEM ALL ??
2258 020552 001006 BNE 1$ ; BR IF NOT.
2259 020554 012737 000020 172516 MOV #20,MMR3 ; YES, SET KT IN 22 BIT MODE...
2260 020562 012737 177400 021252 MOV #177400,IOPAG ;...AND RAISE LIMIT TO 2044K.
2261 020570 005037 010354 1$: CLR BA22
2262 020574 012737 177000 010352 MOV #177000,BA16 ; SET PHYSICAL ADDRESS AT 32K-256 WORDS.
2263 020602 012737 001770 172350 MOV #1770,PAR4 ; SET PAR4 TO THE SAME POINT.
2264 020610 004537 010452 EXMLUP: JSR R5,SETRAN ; RANDOM DATA TO BUF1, ZERO BUF2.
2265 020614 BGNSEG
(3) 020614 104404 TRAP C$BSEG
2266 020616 012701 004364 MOV #BUF1,R1 ; SETUP TO COPY FROM BUF1...
2267 020622 012702 100000 MOV #100000,R2 ;...TO (PAR4)+0
2268 020626 012703 001000 MOV #512,R3
2269 020632 012737 020662 000004 MOV #22$,ERRVEC ; SOONER OR LATER WE'LL TRAP.
2270 020640 005037 010626 CLR TRPFLG
2271 020644 005237 177572 INC MMRO
2272 020650 012122 2$: MOV (R1)+,(R2)+ ;***** KT ON *****
2273 020652 000240 000240 240,240 ; COPY BUFFERS VIA PAR4. *
2274 020656 077304 SOB R3,2$ ; LOOP 'TIL DONE. *
2275 020660 000405 BR 3$ ; *
2276 020662 010237 010626 22$: MOV R2,TRPFLG ; NXM TRAP, SAVE ADDRESS... *
2277 020666 012716 020674 MOV #3$, (SP) ; *
2278 020672 000002 RTI ;...AND CONTINUE. *
2279 020674 005037 177572 3$: CLR MMRO ;***** KT OFF *****
```



```
2280 020700 012737 010624 000004      MOV      #TRAP4,ERRVEC      : RESET TRAP.
2281 020706 004537 010714      JSR      R5,CALCRC         : JUST TO BE CONSISTENT.
2282 020712 000000
2283 020714 004537 010210      JSR      R5,LDFUN
2284 020720 000000 177001      MAINT, -511                : EXECUTE MAINT MODE DMA TO EXT MEM.
2285 020724 004537 010640      JSR      R5,WTCRDY
2286 020730 023727 003402 000003      CMP      R1,TYP,#RLV12X   : BAE IN USE ??
2287 020736 001411      BEQ      33$               : BR IF SO.
2288 020740 013700 003364      MOV      E,CS,R0          : NO, GET EA<17:16> FROM FINAL CSR.
2289 020744 006200      ASR      R0
2290 020746 006200      ASR      R0
2291 020750 006200      ASR      R0
2292 020752 006200      ASR      R0
2293 020754 042700 177774      BIC      #^C3,R0
2294 020760 000402
2295 020762 013700 003376      33$:  MOV      E,BAE,R0        : GET EA<21:16> FROM BAE.
2296 020766 010037 003426      MOV      R0,TMP1          : SAVE FINAL PA IN CASE...
2297 020772 013737 003366 003430      MOV      E,BA,TMP2        : ...WE HAVE A STATUS ERROR.
2298 021000 005737 010626      TST      TRPFLG           : NOW, ARE WE EXPECTING NXM ??
2299 021004 001415      BEQ      4$               : BR IF NOT.
2300 021006 012700 120000      MOV      #ERR!NXM,R0      : YES, DO WE HAVE IT ??
2301 021012 043700 003364      BIC      E,CS,R0
2302 021016 001501      BEQ      8$               : BR IF SO.
2303 021020 004537 007510      JSR      R5,GETERR        : NO, GET WHATEVER'S THERE.
2304 021024 000240      NOP
2305 021026      DFERR      EM90,ERR12    : EXTENDED NXM STATUS WRONG.
(5) 021026      104455
(6) 021030      000232      TRAP      CSERDF
(6) 021032      034147      .WORD     154
(6) 021034      026500      .WORD     EM90
2306 021036      000471      .WORD     ERR12
2307
2308 021040 004537 007510      4$:  JSR      R5,GETERR        : ANY OTHER STATUS ERRORS ??
2309 021044 000405      BR      44$              : BR IF NOT.
2310 021046      DFERR      EM91,ERR13    : STATUS INCORRECT ON EXT MEMORY ACCESS.
(5) 021046      104455      TRAP      CSERDF
(6) 021050      000233      .WORD     155
(6) 021052      034223      .WORD     EM91
(6) 021054      026534      .WORD     ERR13
2311 021056      000461      BR      8$
2312
2313 021060 012701 100000      44$:  MOV      #100000,R1        : POINT TO BUF1 IN EXT MEMORY.
2314 021064 012702 101000      MOV      #101000,R2        : DITTO BUF2.
2315 021070 005003      CLR      R3                : WORD NUMBER.
2316 021072 012704 000377      MOV      #255.,R4          : LOOP COUNT.
2317 021076 005237 177572      INC      MMRO
2318 021102 012137 003432      5$:  MOV      (R1)+,GDDAT
2319 021106 012237 003434      MOV      (R2)+,BDDAT
2320 021112 023737 003432 003434      CMP      GDDAT,BDDAT
2321 021120 001002      BNE      6$
2322 021122 005203      INC      R3
2323 021124 077412      SOB     R4,5$
2324 021126 005037 177572      6$:  CLR      MMRO
2325 021132 005704      TST     R4
2326 021134 001432      BEQ     8$
2327 021136 010337 003424      MOV     R3,TMPO
: LOOP COMPLETED (NO ERRORS) ??
: BR IF SO.
: NO, SAVE FAILING WORD NUMBER.
***** KT ON *****
***** KT OFF *****
```

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 M 7
29 -- RLV11/12 MAINTENANCE, EXTENDED MEMORY ACCESS TEST. PAGE 38-2

SEQ 0090

2328	021142	012703	000006		MOV	#6,R3			
2329	021146	015737	172350	003430	MOV	PAR4,TMP2			
2330	021154	005037	003426		CLR	TMP1			
2331	021160	006337	003430	7\$:	ASL	TMP2	: NOW ASSEMBLE BAD PA,		
2332	021164	006137	003426		ROL	TMP1	: ADJUST PAGE TO BIT 21...		
2333	021170	077305			SOB	R3,7\$			
2334	021172	042702	160000		BIC	#C17777,R2	:...STRIP PAR BITS FROM VA...		
2335	021176	162702	000002		SUB	#2,R2			
2336	021202	060237	003430		ADD	R2,TMP2	:...AND COMBINE THE PIECES.		
2337	021206	005537	003426		ADC	TMP1			
2338	021212				DFERR	EM92,ERR14	: DATA ERROR IN EXTENDED MEM.		
(5)	021212	104455						TRAP	C\$ERDF
(6)	021214	000234						.WORD	156
(6)	021216	034300						.WORD	EM92
(6)	021220	026544						.WORD	ERR14
2339									
2340	021222			8\$:	CKLOOP				
(3)	021222	104406						TRAP	C\$CLP1
2341	021224				ENDSEG				
(3)	021224								
(3)	021224	104405						10000\$:	
2342	021226	062737	020000	010352	ADD	#20000,BA16	: INCR PHYSICAL ADDRESS BY 4K...	TRAP	C\$ESEG
2343	021234	005537	010354		ADC	BA22	:...AND OVERFLOW INTO BAE.		
2344	021240	062737	000200	172350	ADD	#200,PAR4	: INCR PAR4 BY 4K ALSO.		
2345	021246	025727	172350	9\$:	CMP	PAR4,(PC)+	: QUIT AT THE I/O PAGE...		
2346	021252	007400		IOPAG:	7400		:...007400 (124k)...		
2347							:...OR 177400 (2044k).		
2348	021254	103002			BHIS	EXMDUN	: EXIT IF DONE.		
2349	021256	000137	020610		JMP	EXMLUP	: LOOP OTHERWISE.		
2350									
2351	021262	012737	004364	010352	EXMDUN:	MOV	#BUF1,BA16	: RESET BA16 AND 22.	
2352	021270	005037	010354		CLR	BA22			
2353	021274				ENDTST				
(3)	021274							L10054:	
(3)	021274	104401						TRAP	C\$ETST


```
2355 .SBTTL
2356 .SBTTL RLV12 DRIVE INTERFACE TESTS (G5388 TLM REQUIRED).
2357 .SBTTL
2358 021276 STARS
2359 (2) *****
2360 * ALL THESE TESTS REQUIRE THE SERVICES OF THE G5388 TEST LOOP MODULE.
2361 *
2362 .SBTTL 30 -- SYS CLK, PWR OK, DRIVE SELECT, READY, AND ERROR BITS.
2363 *
2364 * 1. TEST SYS CLOCK AND PWR OK FROM THE RLV12.
2365 * 2. TEST DRIVE READY AND DRIVE ERROR TO THE RLV12.
2366 021276 * 3. TEST DRIVE SELECT BITS FROM THE RLV12.
2367 STARS
2368 (2) *****
2369 TLM1: BEGIN.TEST
2370 (4) 021276 T30::
2371 021276 004737 023762 JSR PC,TLMOK ; CHECK FOR V12 AND TLM OK...
2372 021302 102561 BVS 7$ ; ...AND EXIT IF NOT.
2373 021304 012777 000001 162032 MOV #1,@TCSR ; RESET...
2374 021312 005077 162026 CLR @TCSR ; ...AND CLEAR TLM CSR (LO BYTE).
2375 021316 017737 162022 003434 MOV @TCSR,BDDAT ; GET TLM STATUS.
2376 021324 012737 004400 003432 MOV #4400,GDDAT ; EXPECT SYSCLK<11> AND PWROK<8>.
2377 021332 023737 003432 003434 CMP GDDAT,BDDAT
2378 021340 001404 BEQ 1$ ; BR IF STATUS IS RIGHT.
2379 (5) 021342 104455 DFERR EM101,ERR2 ; PWR-OK AND/OR SYS-CLK NOT SET IN TLM.
2380 (6) 021344 000235 TRAP CSERDF
2381 (6) 021346 035007 .WORD 157
2382 (6) 021350 026220 .WORD EM101
2383 021352 017700 161746 1$: MOV @RLCS,RO ; GET RLV12 CSR.
2384 021356 010037 003434 MOV RO,BDDAT ; REC'D RLV12 STATUS...
2385 021362 042700 000001 BIC #DRDY,RO ; ...SHOULD HAVE DRDY CLEAR.
2386 021366 010037 003432 MOV RO,GDDAT
2387 021372 023737 003432 003434 CMP GDDAT,BDDAT
2388 021400 001404 BEQ 2$ ; BR IF SO.
2389 (5) 021402 104455 DFERR EM102,ERR2 ; DRIVE READY NOT CLEAR IN RLV12.
2390 (6) 021404 000236 TRAP CSERDF
2391 (6) 021406 035070 .WORD 158
2392 (6) 021410 026220 .WORD EM102
2393 021412 012777 000200 161724 2$: MOV #200,@TCSR ; SET DRIVE READY IN TLM.
2394 021420 017700 161700 MOV @RLCS,RO ; GET RLV12 AGAIN.
2395 021424 010037 003434 MOV RO,BDDAT
2396 021430 052700 000001 BIS #DRDY,RO ; DRDY SHOULD BE SET NOW.
2397 021434 010037 003432 MOV RO,GDDAT
2398 021440 023737 003432 003434 CMP GDDAT,BDDAT
2399 021446 001404 BEQ 3$ ; BR IF SO.
2400 (5) 021450 104455 DFERR EM102,ERR2 ; DRIVE READY NOT SET IN RLV12.
2401 (6) 021452 000237 TRAP CSERDF
2402 (6) 021454 035070 .WORD 159
2403 (6) 021456 026220 .WORD EM102
2404 021460 012777 000100 161656 3$: MOV #100,@TCSR ; CLEAR DRDY, SET DERR IN TLM.
2405 021466 017700 161632 MOV @RLCS,RO ; ONE MORE TIME.
2406 021472 010037 003434 MOV RO,BDDAT
2407 021476 052700 140000 BIS #ERR!DERR,RO ; SHOULD HAVE THESE ERROR BITS.
```



```

2396 021502 010037 003432
2397 021506 023757 003432 003434      MOV    R0,GDDAT
2398 021514 001404      CMP    GDDAT,BDDAT
2399 021516 104455      BEQ   4$      ; BR IF SO.
(5) 021516 104455      DFERR EM103,ERR2 ; DRIVE ERROR STATUS INCORRECT IN RLV12.
(6) 021520 000240      TRAP  CSERDF
(6) 021522 035145      .WORD 160
(6) 021524 026220      .WORD EM103
2400 021526 005077 161612      4$: CLR   @TCSR      ; CLEAR TLM.
2401 021532 005001      CLR   R1        ; INIT DRIVE SELECT BITS<9:8>...
2402 021534 112737 000060 035235      MOVB  #'0,EM104X ; ...AND ERROR TEXT.
2403 021542 012703 000004      MOV   #4,R3     ; LOOP CONTROL.
2404 021546 012777 000200 161550 5$: MOV   #CRDY,@RLCS ; INIT RLV CSR...
2405 021554 050177 161544      BIS   R1,@RLCS  ; ...INSERT DRIVE BITS <9:8>.
2406 021560 017702 161560      MOV   @TCSR,R2  ; GET TLM CSR.
2407 021564 010257 003434      MOV   R2,BDDAT  ; RECV'D TLM SHOULD ECHO DRIVE...
2408 021570 042702 003000      BIC   #BIT10!BIT9,R2 ; ...SELECTED IN TLM<10:9>.
2409 021574 006301      ASL   R1
2410 021576 050102      BIS   R1,R2     ; SET EXPECTED VALUE.
2411 021600 006201      ASR   R1
2412 021602 010237 003432      MOV   R2,GDDAT
2413 021606 023757 003432 003434      CMP   GDDAT,BDDAT
2414 021614 001404      BEQ   6$
2415 021616 104455      DFERR EM104,ERR2 ; DRIVE SELECT FAILS IN TLM.
(5) 021616 104455      TRAP  CSERDF
(6) 021620 000241      .WORD 161
(6) 021622 035227      .WORD EM104
(6) 021624 026220      .WORD ERR2
2416 021626 062701 000400 6$: ADD   #DS1,R1   ; BUMP DRIVE SELECT NUMBER...
2417 021632 105237 035235      INCB  EM104X    ; ...AND THE ERROR TEXT.
2418 021636 077335      SOB   R3,5$
2419 021640 012777 000200 161456 7$: MOV   #CRDY,@RLCS ; RESET DRIVE NUMBER IN RL.
2420 021646
(5) 021646
(3) 021646 104401      L10055: TRAP  CSETST

```

2422
2423
2424 021650
(2)
2425
2426
2427 021650
(2)
2428 021650
(4)
2429 021650 004737 023762
2430 021654 102476
2431 021656 004737 024132
2432 021662 005037 003436
2433 021666 012765 125252 000004
2434 021674 012777 000204 161422
2435 021702 012777 000013 161420
2436 021710 004537 010016
2437 021714 042777 000200 161402
2438 021722 005000
2439 021724 011501 1\$:
2440 021726 100401
2441 021730 077003
2442 021732 016537 000002 003434 2\$:
2443 021740 004537 010640
2444 021744 012737 000013 003432
2445 021752 023737 003432 003434
2446 021760 001404
2447 021762
(5) 021762 104455
(6) 021764 000242
(6) 021766 035311
(6) 021770 026304
2448 021772 013737 003372 003434 3\$:
2449 022000 012737 125252 003432
2450 022006 023737 003432 003434
2451 022014 001404
2452 022016
(5) 022016 104455
(6) 022020 000243
(6) 022022 035373
(6) 022024 026304
2453 022026 004537 007510 4\$:
2454 022032 000407
2455 022034 012737 036670 003426
2456 022042
(5) 022042 104455
(6) 022044 000244
(6) 022046 035443
(6) 022050 026666
2457 022052 5\$:
(3) 022052
(3) 022052 104401

.SBTTL 31 -- DRIVE COMMAND, STATUS AND STATUS CLOCK.

STARS

: * NOW TEST DRIVE COMMAND, STATUS, AND STATUS CLOCK
: * USING A "GET STATUS" COMMAND.

STARS

TLM2: BEGIN.TEST

T31::

```
JSR PC, TLMOK  
BVS 5$ : EXIT IF NO GOOD.  
JSR PC, CLRTLM : RESET TLM, AND SET DRDY.  
CLR INIMP  
MOV #125252, 4(R5) : SET PHONEY STATUS WORD.  
MOV #CRDY!GSTAT, @RLCS : SET "GET STATUS" COMMAND...  
MOV #DRST!GSBIT!MK, @RLDA ; ...AND RESET STATUS BITS.  
JSR R5, BEFORE  
BIC #CRDY, @RLCS : XCT IT.  
CLR R0  
MOV (R5), R1 : GET TCSR...  
BMI 2$ : ...PROCEED WHEN "NEW SKGS" SETS...  
SOB R0, 1$ : ...BUT DON'T WAIT FOREVER !!!  
MOV 2(R5), BDDAT : SAVE RECV'D DRIVE COMMAND (SKGS)...  
JSR R5, WTCRDY  
MOV #DRST!GSBIT!MK, GDDAT ; ...SHOULD LOOK LIKE THIS.  
CMP GDDAT, BDDAT  
BEQ 3$ : BR IF STATUS COMMAND WAS RIGHT.  
DFERR EM105, ERR4 : GET STATUS COMMAND WRONG IN TLM.  
TRAP .WORD CSERDF  
TRAP .WORD 162  
TRAP .WORD EM105  
TRAP .WORD ERR4  
MOV E.MP, BDDAT : RETURNED DRIVE STATUS...  
MOV #125252, GDDAT : ...SHOULD = PHONEY.  
CMP GDDAT, BDDAT  
BEQ 4$ : BR IF RETURNED STATUS IS RIGHT.  
DFERR EM106, ERR4 : RETURNED STATUS INCORRECT IN RLMP.  
TRAP .WORD CSERDF  
TRAP .WORD 163  
TRAP .WORD EM106  
TRAP .WORD ERR4  
JSR R5, GETERR : FINAL CONTROLLER STATUS...  
BR 5$ : ...SHOULD BE ERROR FREE.  
MOV #EXPNON, TMP1  
DFERR EM107, ERR20A : RLV12 ERRORS AFTER RESET GET STATUS  
TRAP .WORD CSERDF  
TRAP .WORD 164  
TRAP .WORD EM107  
TRAP .WORD ERR20A  
L10056: TRAP CSETST
```

2459
2460
2461 022054
(2)
2462
2463 022054
(2)
2464 022054
(4)
2465 022054
2466 022060
2467 022062
2468 022066
2469 022072
2470 022076
2471 022104
2472 022112
2473 022116
2474 022124
2475 022126
2476 022130
2477 022132
2478 022134
2479 022142
2480 022146
2481 022154
2482 022162
2483 022164
(5) 022164
(6) 022166
(6) 022170
(6) 022172
2484 022174
2485 022200
2486 022202
2487 022210
(5) 022210
(6) 022212
(6) 022214
(6) 022216
2488 022220
(3) 022220
(3) 022220

004737 023762
102457
004737 024132
005037 003436
052715 000024
012777 000206 161220
012777 077601 161216
004537 010016
042777 000200 161200
005000
011501
100401
077003
016537 000002 003434
004537 010640
012737 077601 003432
023737 003432 003434
001404
104455
000245
035516
026304
004537 007510
000407
012737 036670 003426
104455
000246
035577
026666
104401

```
.SBTTL 32 -- DRIVE COMMAND, SEEK DIFF AND SECTOR PULSE.  
STARS  
:*****  
: * TEST SECTOR PULSE USING A "SEEK" COMMAND.  
STARS  
:*****  
TLM3: BEGIN.TEST  
T32::  
JSR PC,TLMOK  
BVS 4$ : BR IF NOT OK.  
JSR PC,CLRTL : RESET TTL, SET DRDY.  
CLR INIMP  
BIS #24,(R5) : ENABLE SECTOR GENERATOR (NOM CLOCK).  
MOV #CRDY!SEEK,@RLCS : SET "SEEK" COMMAND...  
JSR R5,BEFORE :...AND A PHONEY SEEK DIFFERENCE.  
BIC #CRDY,@RLCS : XCT IT.  
CLR R0  
1$: MOV (R5),R1  
BMI 2$ : PROCEED WHEN "NEW SKGS" SETS...  
SOB R0,1$ :...KEEP-ALIVE !!  
2$: MOV 2(R5),BDDAT : SAVE RECV'D DRIVE COMMAND...  
JSR R5,WTCDY :...SHOULD = PHONEY DIFFERENCE.  
MOV #77600!MK,GDDAT  
CMP GDDAT,BDDAT  
BEQ 3$ : BR IF SO.  
DFERR EM110,ERR4 : DRIVE COMMAND WRONG IN TLM.  
TRAP .WORD CSERDF  
TRAP .WORD 165  
TRAP .WORD EM110  
TRAP .WORD ERR4  
3$: JSR R5,GETERR : FINAL CONTROLLER STATUS...  
BR 4$ :...SHOULD BE ERROR FREE.  
MOV #EXPNON,TMP1  
DFERR EM111,ERR20A : RLV ERRORS AFTER SEEK COMMAND  
TRAP .WORD CSERDF  
TRAP .WORD 166  
TRAP .WORD EM111  
TRAP .WORD ERR20A  
4$: ENDTST  
L10057:  
TRAP CSETST
```


2490
2491
2492 022222
(2)
2493
2494 022222
(2)
2495 022222
(4)
2496 022222 004737 023762
2497 022222 102524
2498 022230 004737 024132
2499 022234 052715 000024
2500 022240 004537 010160
2501 022244 000002
2502 022246 012777 000212 161050
2503 022254 012737 177600 003436
2504 022262 013777 003436 161042
2505 022270 012777 004364 161030
2506 022276 012777 000205 161024
2507 022280 004537 010016
2508 0222810 042777 000200 161006
2509 0222816 005000
2510 0222820 011501
2511 0222822 010137 003434 18:
2512 0222826 052701 030000
2513 0222832 010137 003432
2514 0222836 023737 003432 003434
2515 0222844 001410
2516 0222846 077014
2517 0222850 004537 010640
2518 0222854
(5) 0222854 104455
(6) 0222856 000247
(6) 0222860 035644
(6) 0222862 026304
2519 0222864 000402
2520 0222866 004537 010640 28:
2521 0222872 011537 003434
2522 0222876 013737 003434 003432
2523 0222880 042737 060000 003432
2524 0222884 023737 003432 003434
2525 0222890 001415
2526 0222892 032737 020000 003434
2527 0222896 001405
2528 0222898
(5) 0222898 104455
(6) 0222902 000250
(6) 0222906 036010
(6) 0222910 026304
2529 0222914 000404
2530 0222918
(5) 0222918 104455
(6) 0222922 000251
(6) 0222926 035736
(6) 0222930 026304

.SBTTL 33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.
STARS

* TEST THE WRITE GATE, WRITE DATA ACTIVE, AND WRITE GATE ERROR.
STARS

TLM4: BEGIN.TEST
T33::
JSR PC, TLMOK
BVS 5\$: BR IF NOT OK.
JSR PC, CLRTLM : RESET TLM, SET DRDY.
BIS #24, (R5) : ENABLE SECTOR GEN (NOM CLOCK).
JSR R5, WDELAY : DELAY SO THAT WE ENTER THE TLM...
2 : ...SEQUENCE ASYNCHRONOUSLY.
MOV #CRDY!WRITE, @RLCS ; SET COMMAND TO WRITE...
MOV #-128, INIMP
MOV INIMP, @RLMP : ...128 WORDS...
MOV #BUF1, @RLBA : ...FROM BUF1...
MOV #205, @RLDA : ...TO CYL 1, SECT 5.
JSR R5, BEFORE
BIC #CRDY, @RLCS : XCT IT.
C'R R0
MOV (R5), R1 : TLM STATUS => R1.
MOV R1, BDDAT : RECV'D TLM STATUS...
BIS #BIT13!BIT12, R1 : ...WILL GET WRITE GATE AND DATA ACTIVE.
MOV R1, GDDAT
CMP GDDAT, BDDAT
BEQ 2\$: BR IF AND WHEN THAT OCCURS...
SOB R0, 18 : ...WITHIN REASON -- OF COURSE.
JSR R5, WTCRDY
DFERR EM112, ERR4 : WRITE GATE AND/OR DATA ACTIVE NOT SET.
TRAP CSERDF
.WORD 167
.WORD EM112
.WORD ERR4
SKP2
JSR R5, WTCRDY
MOV (R5), BDDAT : FINAL TLM STATUS...
MOV BDDAT, GDDAT
BIC #BIT14!BIT13, GDDAT ; ...ERROR AND GATE SHOULD BE CLEAR.
CMP GDDAT, BDDAT
BEQ 4\$: BR IF SO.
BIT #BIT13, BDDAT : IT'S NOT, IS GATE STILL SET ??
BEQ 3\$
DFERR EM113A, ERR4 : WRITE GATE STILL SET AFTER DONE.
TRAP CSERDF
.WORD 168
.WORD EM113A
.WORD ERR4
BR 4\$
DFERR EM113, ERR4 : WRITE GATE ERROR IN TLM.
TRAP CSERDF
.WORD 169
.WORD EM113
.WORD ERR4

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 F 8 PAGE 42-1
33 -- WRITE GATE, WRITE GATE ERROR, AND WRITE DATA.

SEQ 0096

2531 022454 004537 007510
2532 022460 000407
2533 022462 012737 036670 003426
2534 022470
(5) 022470 104455
(6) 022472 000252
(6) 022472 036066
(6) 022476 026666
2535 022500
(3) 022500
(3) 022500 104401

4\$: JSR R5,GETERR ; FINAL RLV STATUS...
BR 5\$;...SHOULD BE ERROR FREE.
MOV #EXPNON,TMP1
DFERR EM114,ERR20A ; RL ERROR AFTER WRITE.

TRAP CSERDF
.WORD 170
.WORD EM114
.WORD ERR20A

5\$: ENDTST

L10060: TRAP CSETST

2537
2538
2539 022502
(2)
2540
2541
2542
2543
2544
2545 022502
(2)
2546 022502
(4)
2547 022502 004737 023766
2548 022506 102525
2549 022510 012737 177600 003436
2550 022516 013737 003314 022544
2551 022524 042737 177770 022544
2552 022532 001002
2553 022534 005237 022544
2554 022540 004737 024214
2555 022544 000001
2556 022546 102010
2557 022550 113737 036532 036521
2558 022556
(5) 022556 104455
(6) 022560 000253
(6) 022562 036454
(6) 022564 026160
2559 022566 000465
2560 022570 032737 000004 003312 1\$:
2561 022576 001006
2562 022600 012715 000224
2563 022604 012737 036536 003430
2564 022612 000423
2565 022614 032737 000002 003312 2\$:
2566 022622 001006
2567 022624 012715 000230
2568 022630 012737 036554 003430
2569 022636 000411
2570 022640 032737 000001 003312 3\$:
2571 022646 001035
2572 022650 012715 000220
2573 022654 012737 036567 003430
2574
2575 022662 032737 040000 003312 4\$:
2576 022670 001004
2577 022672 105037 036602
2578 022676 004737 022766
2579 022702 005737 003312 5\$:
2580 022706 100407
2581 022710 052715 000040
2582 022714 112737 000040 036602
2583 022722 004737 022766
2584
2585 022726 032715 000004 6\$:

.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
T34::
JSR PC,TLMPID : VERIFY TLM OK AND PROM ID.
BVS 8\$: EXIT IF NOT OK.
MOV #-128,INIMP : INIT WORD COUNT.
MOV SNGLSEC,12\$: GET SECTOR NUMBER (IF ANY).
BIC #^C7,12\$
BNE 11\$: NZ = SINGLE SECTOR MODE.
INC 12\$: OTHERWISE, USE ALL STARTING WITH 1.
JSR PC,PSX : SET BUFFERS AND DISK ADDRESS...
1 : ...FOR THIS SECTOR NUMBER.
BVC 1\$: BR IF SECTOR FOUND.
MOVB TSECX,EM119X : SET ASCII SECTOR NUMBER.
DFERR EM119,NOSIG : CAN'T FIND SECTOR.
TRAP CSERDF
.WORD 171
.WORD EM119
.WORD NOSIG
BR 7\$
BIT #BIT2,MPXCLK : NOMINAL CLOCK ENABLED ??
BNE 2\$: BR IF NOT.
MOV #224,(R5) : YES, SET NOMINAL CLOCK, NO PEAK SHIFT.
MOV #TCLK0,TMP2
BR 4\$
BIT #BIT1,MPXCLK : FAST CLOCK ENABLED ??
BNE 3\$: BR IF NOT.
MOV #230,(R5) : YES, SET FAST CLOCK, NO PEAK SHIFT.
MOV #TCLK1,TMP2
BR 4\$
BIT #BIT0,MPXCLK : SLOW CLOCK ENABLED ??
BNE 7\$: BR IF NOT.
MOV #220,(R5) : YES, SET SLOW CLOCK, NO PEAK SHIFT.
MOV #TCLK2,TMP2
BIT #BIT14,MPXCLK : MIN PEAK SHIFT ENABLED ??
BNE 5\$: BR IF NOT.
CLRB TPEAK : YES, PEAK SHIFT IS OFF...
JSR PC,READSEC : ...READ SECTOR AND VERIFY DATA.
TST MPXCLK : MAX PEAK SHIFT ENABLED ??
BMI 6\$: BR IF NOT.
BIS #40,(R5) : YES, MAX PEAK SHIFT ON...
MOVB #40,TPEAK : ...EXTEND CLOCK MESSAGE...
JSR PC,READSEC : ...DO IT ALL AGAIN.
BIT #4,(R5) : NOMINAL DONE ??


```
2586 022732 001330      BNE      2$          : LOOP TO 'FAST' IF SO.
2587 022734 032715 000010 BIT      #10,(R5)   : FAST DONE ??
2588 022740 001337      BNE      3$          : LOOP TO 'SLOW' IF SO.
2589                                     : ...OTHERWISE, THIS SECTOR IS DONE.
2590
2591 022742 032737 000007 003314 7$:  BIT      #7,SNGLSEC : SINGLE SECTOR MODE ??
2592 022750 001004      BNE      8$          : WE'RE DONE IF SO.
2593 022752 023727 022544 000006 CMP      12$,#6     : NO, LAST SECTOR DONE ??
2594 022760 103665      BLO     10$         : GO 'ROUND AGAIN IF NOT...
2595 022762      8$:  EXIT    TST          : ...OTHERWISE, NEXT TEST.
(3) 022762 104432                                     TRAP      CSEXIT
(3) 022764 000376                                     .WORD    L10061-.
2596
2597                                     : SUBROUTINE TO READ DATA FROM TLM TO BUF2 AND VERIFY (AGAINST BUF1).
2598                                     : DISK ADDRESS AND EXPECTED DATA (BUF1) MUST BE PRESET BEFORE CALL.
2599                                     : WORD COUNT (-128.) IS PRESET FOR ALL.
2600
2601 022766      READSEC: BGNSEG
(3) 022766 104404                                     TRAP      CSBSEG
2602 022770 012700 005364      MOV      #BUF2,R0
2603 022774 012701 000200      MOV      #128,R1
2604 023000 012720 177777      1$:  MOV      #-1,(R0)+ : CLEAR RECEIVING BUFFER.
2605 023004 077103      SOB      R1,1$
2606 023006 105037 036205      CLRB    EM115X     : INIT ERROR TEXT FOR 'READ'.
2607 023012 105037 036355      CLRB    EM117X
2608 023016 012777 000214 160300 MOV      #CRDY!READ,ARLCS : SET COMMAND TO READ...
2609 023024 013777 003436 160300 MOV      INIMP,ARLMP  : ...128 WORDS...
2610 023032 013777 003440 160270 MOV      INIDA,ARLDA  : ...FROM CYL X, SECT Y...
2611 023040 012777 005364 160260 MOV      #BUF2,ARLBA  : ...TO BUF2.
2612 023046 004537 010016      2$:  JSR      R5,BEFORE
2613 023052 042777 000200 160244 BIC      #CRDY,ARLCS : XCT.
2614 023060 004537 010640      JSR      R5,WTCRDY  : WAIT FOR RL DONE.
2615 023064 123737 024506 003365 CMPB     PSERR,E.CS+1 : FINAL STATUS RIGHT ??
2616 023072 001412      BEQ     3$          : BR IF SO.
2617 023074 004537 007510      JSR      R5,GETERR  : NO, GET WHATEVER'S THERE.
2618 023100      NOP
2619 023102 013737 024504 003426 MOV      PSETXT,TMP1
2620 023110      DFERR   EM115,ERR20 : ERROR STATUS WRONG ON READ.
(5) 023110 104455                                     TRAP      CSERDF
(6) 023112 000254                                     .WORD    172
(6) 023114 036137                                     .WORD    EM115
(6) 023116 026662                                     .WORD    ERR20
2621 023120 023727 024504 036624 3$:  CMP      PSETXT,#EXPHCRC : NOW IS THIS THE BAD HEADER SECTOR ??
2622 023126 001062      BNE     CHKDATA    : BR IF NOT.
2623
2624 023130 012700 000007      READNH: MOV      #7,R0          : YES IT IS, SET UP TO READ HEADERS.
2625 023134 112737 000040 036205 MOVB     #40,EM115X   : EXTEND ERROR TEXT TO 'READ NH'
2626 023142 112737 000040 036355 MOVB     #40,EM117X
2627 023150 012777 000210 160146 1$:  MOV      #CRDY!RDHDR,ARLCS
2628 023156 042777 000200 160140 BIC      #CRDY,ARLCS : XCT READ HEADER...
2629 023164 105777 160134      2$:  TSTB    ARLCS      : ...AND WAIT FOR DONE.
2630 023170 100375      BPL     2$
2631 023172 027727 160134 000206 CMP      ARLMP,#<1_7.>!6 : FOUND BAD SECTOR-1 ??
2632 023200      BEQ     3$          : BR IF SO.
2633 023202      SOB     R0,1$
2634 023204      DFERR   EM116,ERR21 : CAN'T FIND CYL 1, SECT 6
```

```
(5) 023204 104455
(6) 023206 000255 TRAP CSEDF
(6) 023210 036233 .WORD 173
(6) 023212 026742 .WORD EM116
2635 023214 000460 .WORD ERR21
2636 023216 012777 000216 160100 3$: BR READDUN ; ABORT THE REST.
2637 023220 013777 003436 160100 MOV #CRDY!RDNHDR,@RLCS ; OK, SET READ-NO-HEADER COMMAND...
2638 023222 004537 010016 JSR R5,BEFORE ;...INSURE WORD COUNT IS STILL THERE...
2639 023226 042777 000200 160060 BIC #CRDY,@RLCS ;...AND XCT IT.
2640 023228 004537 010640 JSR R5,WTCDY
2641 023230 004537 007510 JSR R5,GETERR ; SHOULD HAVE NO ERRORS...
2642 023232 000407 BR CHKDATA ;...BR IF SO.
2643 023236 012737 036670 003426 MOV #EXPNON,TMP1
2644 023238 000407 DFERR EM115,ERR20 ; ERROR STATUS INCORRECT AFTER READ NH
(5) 023264 104455 TRAP CSEDF
(6) 023266 000256 .WORD 174
(6) 023270 036137 .WORD EM115
(6) 023272 026662 .WORD ERR20
2645
2646 023274 012700 004444 CHKDATA: MOV #BUF1+48.,R0 ; EXPD DATA FIELD BEGINS AT WORD 24...
2647 023300 012701 005364 MOV #BUF2,R1 ;...RECVD DATA AT WORD 0.
2648 023304 005002 CLR R2 ;...AND WORD NUMBER.
2649 023306 021011 1$: CMP (R0),(R1) ; COMPARE EXP'D VS REC'D DATA...
2650 023310 001415 BEQ 2$ ; AND BR IF OK.
2651 023312 011037 003432 MOV (R0),GDDAT ; IT'S NOT, SET EXP'D...
2652 023316 011137 003434 MOV (R1),BDDAT ;...REC'D...
2653 023320 010137 003424 MOV R1,TMP0 ;...ADDRESS...
2654 023322 010237 003426 MOV R2,TMP1 ;...AND WORD NUMBER.
2655 023326 010237 DFERR EM117,ERR22 ; DATA INCORRECT ON READ OR READ NH.
(5) 023332 104455 TRAP CSEDF
(6) 023334 000257 .WORD 175
(6) 023336 036313 .WORD EM117
(6) 023340 026752 .WORD ERR22
2656 023342 000405
2657 023344 022021 2$: BR 3$ ; BUMP ADDRESSES...
2658 023346 005202 INC R2 ;...AND WORD NUMBER.
2659 023350 032702 000177 BIT #177,R2
2660 023354 001354 BNE 1$ ;...AND LOOP 'TIL DONE.
2661 023356
2662 023358 READDUN: ENDSEG
(3) 023356 104405 10000$: TRAP CSESEG
2663 023360 000207 RTS PC
2664 023362
(3) 023364
(3) 023366 104401 L10061: TRAP CSETST
```



```
.SBTTL 35 -- WRITE CHECK.
STARS
*****
* DO A WRITE CHECK USING SECTOR AND CLOCK OPTIONS AS BEFORE.
STARS
*****
fLM6: BEGIN.TEST
T35::
2666
2667
2668 023364
2669
2670 023364
2671 023364
2672 023364 004737 023766 JSR PC,TLMPID
2673 023370 102525 BVS 8$ ; EXIT IF NOT OK.
2674 023372 012737 177600 003436 MOV #-128,INIMP ; INIT WORD COUNT.
2675 023400 013737 003314 023426 MOV SNGLSEC,12$ ; GET SECTOR NUMBER (IF ANY).
2676 023406 042737 177770 023426 BIC #^C7,12$
2677 023414 001002 BNE 11$ ; NZ = SINGLE SECTOR MODE.
2678 023416 005237 023426 10$: INC 12$ ; OTHERWISE, USE ALL STARTING WITH 1.
2679 023422 004737 024214 11$: JSR PC,PSX ; SET BUFFERS AND DISK ADDRESS...
2680 023426 000001 12$: 1 ; ...FOR THIS SECTOR NUMBER.
2681 023430 102010 BVC 1$
2682 023432 113737 036532 036521 MOVB TSECX,EM119X ; SET ASCII SECTOR NUMBER.
2683 023440 DFERR EM119,NOSIG ; CAN'T FIND SECTOR.
2684 023440 104455
2685 023442 000260 TRAP CSERDF
2686 023444 036454 .WORD 176
2687 023446 026160 .WORD EM119
2688 023450 000465 .WORD NOSIG
2689 023452 032737 000004 003312 1$: BR 7$
2690 023460 001006 BIT #BIT2,MPXCLK ; NOMINAL CLOCK ENABLED ??
2691 023462 012715 000224 BNE 2$ ; BR IF NOT.
2692 023466 012737 036536 003430 MOV #224,(R5) ; YES, SET NOMINAL CLOCK, NO PEAK SHIFT.
2693 023474 000423 MOV #TCLK0,TMP2
2694 023476 032737 000002 003312 2$: BR 4$
2695 023504 001006 BIT #BIT1,MPXCLK ; FAST CLOCK ENABLED ??
2696 023506 012715 000230 BNE 3$ ; BR IF NOT.
2697 023512 012737 036554 003430 MOV #230,(R5) ; YES, SET FAST CLOCK, NO PEAK SHIFT.
2698 023520 000411 MOV #TCLK1,TMP2
2699 023522 032737 000001 003312 3$: BR 4$
2700 023530 001035 BIT #BIT0,MPXCLK ; SLOW CLOCK ENABLED ??
2701 023532 012715 000220 BNE 7$ ; BR IF NOT.
2702 023536 012737 036567 003430 MOV #220,(R5) ; YES, SET SLOW CLOCK, NO PEAK SHIFT.
2703 023544 032737 040000 003312 4$: MOV #TCLK2,TMP2
2704 023552 001004 BIT #BIT14,MPXCLK ; MIN PEAK ENABLED ??
2705 023554 105037 036602 BNE 5$ ; BR IF NOT.
2706 023560 004737 023650 CLR B TPEAK ; YES, PEAK SHIFT IS OFF...
2707 023564 005737 003312 JSR PC,WRITCHK ; ...WRITE-CHECK THIS SECTOR.
2708 023570 100407 5$: TST MPXCLK ; MAX PEAK SHIFT ENABLED ??
2709 023572 052715 000040 BMI 6$ ; BR IF NOT.
2710 023576 112737 000040 036602 BIS #40,(R5) ; YES, MAX PEAK SHIFT ON...
2711 023604 004737 023650 MOVB #40,TPEAK ; ...EXTEND CLOCK MESSAGE...
2712 023610 032715 000004 6$: JSR PC,WRITCHK ; ...AND DO IT AGAIN.
2713 023614 001330 6$: BIT #4,(R5) ; NOMINAL DONE ??
2714 023616 032715 000010 BNE 2$ ; LOOP TO "FAST" IF SO.
2715 023622 001337 BIT #10,(R5) ; FAST DONE ??
2716 BNE 3$ ; LOOP TO "SLOW" IF SO.
2717 ; ...OTHERWISE, THIS SECTOR IS DONE.
```



```
2715 023624 032737 000007 003314 7$: BIT #7,SNGLSEC : SINGLE SECTOR MODE ??
2716 023632 001004 : BNE 8$ : WE'RE ALL DONE IF SO.
2717 023634 023727 023426 000006 : CMP 12$,#6 : NO, LAST SECTOR DONE ??
2718 023642 103665 : BLO 10$ : GO 'ROUND AGAIN IF NOT...
2719 023644 : 8$: EXIT TST : ...OTHERWISE, WE'RE ALL DONE.
(3) 023644 104432 : TRAP CSEXIT
(3) 023646 000112 : .WORD L10062-.
2721 :
2722 : SUBROUTINE TO WRITE CHECK A SECTOR AGAINST THE DATA IN 'BUF1'.
2723 :
2724 023650 WRITCHK: BGNSEG
(3) 023650 104404 : TRAP CSBSEG
2725 023652 012777 000202 157444 : MOV #CRDY!WRCHK,@RLCS ; SET COMMAND TO WRITE-CHECK...
2726 023660 013777 003436 157444 : MOV INIMP,@RLMP : ...128 WORDS...
2727 023666 013777 003440 157434 : MOV INIDA,@RLDA : ...FROM CYL X, SECT Y...
2728 023674 012777 004444 157424 : MOV #BUF1+48,@RLBA : ...AGAINST BUF1.
2729 023702 004537 010016 : JSR R5,BEFORE
2730 023706 042777 000200 157410 : BIC #CRDY,@RLCS : XCT.
2731 023714 004537 010640 : JSR R5,WTCDY : WAIT FOR RL DONE.
2732 023720 123737 024506 003365 : CMPB PSERR,E.CS+1 : FINAL STATUS RIGHT ??
2733 023726 001412 : BEQ 1$ : BR IF SO.
2734 023730 004537 007510 : JSR R5,GETERR : NO, GET WHATEVER'S THERE.
2735 023734 000240 : NOP
2736 023736 013737 024504 003426 : MOV PSETXT,TMP1
2737 023744 : DFERR EM118,ERR20 : ERROR STATUS WRONG ON WRITE-CHECK
(5) 023744 104455 : TRAP CSERDF
(6) 023746 000261 : .WORD 177
(6) 023750 036403 : .WORD EM118
(6) 023752 026662 : .WORD ERR20
2738 023754 1$: ENDSEG
(3) 023754 :
(3) 023754 104405 : 10000$: TRAP CSESEG
2739 023756 000207 : RTS PC
2740 023760 :
(3) 023760 :
(3) 023760 104401 : L10062: TRAP CSETST
```

```
2742  
2743  
2744  
2745  
2746  
2747 023762 005000  
2748 023764 000401  
2749 023766 005200  
2750 023770 013737 003344 003432  
2751 023776 001453  
2752 024000 005037 010626  
2753 024004 005077 157334  
2754 024010 000240  
2755 024012 005737 010626  
2756 024016 001025  
2757 024020 012737 004400 003432  
2758 024026 017737 157312 003434  
2759 024034 001423  
2760 024036 005700  
2761 024040 001413  
2762 024042 004737 024132  
2763 024046 010037 003434  
2764 024052 013737 003310 003432  
2765 024060 023737 003432 003434  
2766 024066 001013  
2767 024070 000207  
2768  
2769 024072  
(5) 024072 104455  
(6) 024074 000262  
(6) 024076 034574  
(6) 024100 026170  
2770 024102 000411  
2771 024104  
(5) 024104 104455  
(6) 024106 000263  
(6) 024110 034630  
(6) 024112 026220  
2772 024114 000404  
2773 024116  
(5) 024116 104455  
(6) 024120 000264  
(6) 024122 034713  
(6) 024124 026220  
2774 024126 000262  
2775 024130 000207  
2776  
2777  
2778  
2779  
2780  
2781  
2782 024132 013705 003344  
2783 024136 012715 000701  
2784 024142 016500 000006  
2785 024146 005065 000002
```

...
: SUBROUTINE TO SEE IF TLM IS THERE AND PROPERLY CABLED TO RLV12.
: ALSO CHECK THAT CORRECT PROM IS INSTALLED.
: REPORT ERROR AND RETURN WITH 'V' SET IF NOT.
: ...
TLMOK: CLR R0 ; DISABLE ID CHECK.
SKP1
TLMPID: INC R0 ; ENABLE ID CHECK.
MOV TCSR,GDDAT ; GET CSR POINTER.
BEQ 4\$; ABORT IF IT'S NOT DEFINED.
CLR TRPFLG
CLR @TCSR ; TLM ARE YOU REALLY THERE ??
240
TST TRPFLG
BNE 1\$; ERROR IF NOT.
MOV #4400,GDDAT ; OTHERWISE
MOV @TCSR,BDDAT ; ...GET INITIAL STATUS.
BEQ 2\$; IF ZERO, CABLE ISN'T CONNECTED.
TST R0 ; ID VERIFICATION REQUIRED ??
BEQ 5\$; WE'RE DONE IF NOT.
JSR PC,CLRTLM ; YES, RESET, ID => R0.
MOV R0,BDDAT
MOV PROMID,GDDAT ; GET EXPECTED ID.
CMP GDDAT,BDDAT
BNE 3\$; BR IF ID INCORRECT.
RTS PC ; RETURN, 'V' = 0.
5\$:
1\$: DFERR EM100,ERR1 ; TIME-OUT ON TLM ADDRESS.
TRAP .WORD CSERDF
178
2\$: BR 4\$
DFERR EM100A,ERR2 ; CABLE PROBABLY NOT INSTALLED.
TRAP .WORD EM100
179
3\$: BR 4\$
DFERR EM100B,ERR2 ; PROM ID INCORRECT.
TRAP .WORD EM100B
180
4\$: SEV PC ; SET 'ERROR' FLAG...
RTS PC ; ...AND RETURN.
TRAP .WORD ERR2
181

...
: SUBROUTINE TO RESET TEST LOOP MODULE.
: RETURN WITH 'DRIVE READY' SET, TLM PROM ID IN R0,
: AND TLM CSR ADDRESS IN R5.
: AS LONG AS WE'RE HERE, CLEAR RLBA, RLMP, AND RLBAE AS WELL.
: ...
CLRTLM: MOV TCSR,R5 ; DEDICATE R5 TO THE TLM.
MOV #1,(R5) ; RESET...
MOV 6(R5),R0 ; ...PROM ID => R0.
CLR 2(R5) ; CLEAR SKGS...

```

2786 024152 005065 000004 CLR 4(R5) ;...AND PSUEDO-STATUS REGISTERS.
2787 024156 012715 000002 MOV #2,(R5) ;...CLEAR FLAGS ??
2788 024162 012715 000200 MOV #200,(R5) ;...SET DRIVE READY.
2789 024166 005077 157134 CLR @RLBA
2790 024172 005077 157134 CLR @RLMP
2791 024176 023727 003402 000003 CMP R1YP,#RLV12X
2792 024204 001002 BNE 1$
2793 024206 005077 157122 CLR @RLBAE
2794 024212 000207 1$: RTS PC
2795
2796 ; SUBROUTINE TO SET UP BUFFERS AND DISK ADDRESSES
2797 ; FOR READING EACH OF THE 6 PSUEDO SECTORS.
2798 ; USE HEADER TABLE AND SEARCH ROUTINE IN THE DUMP UTILITY.
2799 ; IF SUCCESSFUL, CONTENTS OF SELECTED SECTOR (160. WORDS) ARE IN 'BUF1'.
2800 ; CALL: JSR PC,PSX
2801 ; N ; SECTOR NUMBER,
2802 ; BVC XX ; ON RETURN, 'V' IS SET IN SECTOR NOT FOUND.
2803 ; ERROR
2804 ; XX: ; CONTINUE.
2805
2806 024214 017600 000000 PSX: MOV @ (SP),RO ; GET SECTOR NUMBER (1 - 6)...
2807 024220 005300 DEC RO ;...MAKE IT (0 - 5)
2808 024222 006300 ASL RO ;...SHIFT UP TO AN INDEX.
2809 024224 062716 000002 ADD #2,(SP) ; ADJUST RETURN PC...
2810 024230 000170 024234 JMP @1$(RO) ;...AND DO ONE OF THE FOLLOWING:
2811 024234 024252 1$: PS1
2812 024236 024270 PS2
2813 024240 024322 PS3
2814 024242 024354 PS4
2815 024244 024372 PS5
2816 024246 024442 PS6
2817 024250 024410 PS7
2818 ; SOONER OR LATER SOME TURKEY WILL...
2819 ; ...TRY TO ACCESS SECTOR 7.
2820
2821 024252 112737 000061 036532 PS1: .ENABL LSB
2822 024260 012737 000205 003440 MOVB #1,TSECX ; TLM SECTOR 1...
2823 024266 000473 BR #<1_7.>!5,INIDA ;...IS CYL 1 SECT 5...
2824 ;...AND IS ERROR FREE.
2825 024270 112737 000062 036532 PS2: MOVB #2,TSECX ; TLM SECTOR 2
2826 024276 012737 000206 003440 MOV #<1_7.>!6,INIDA ;...IS CYL 1, SECT 6...
2827 024304 012737 000210 024506 MOV #^H^ZERR!D^C^R^C>,PSERR
2828 024312 012737 036650 024504 MOV #EXPDCRC,PSETXT ;...AND HAS A BAD DATA CRC.
2829 024320 000463 BR 2$
2830
2831 024322 112737 000063 036532 PS3: MOVB #3,TSECX ; TLM SECTOR 3
2832 024330 012737 000207 003440 MOV #<1_7.>!7,INIDA ;...IS CYL 1, SECT 7...
2833 024336 012737 000214 024506 MOV #^H^ZERR!H^C^R^C!O^P^I>,PSERR
2834 024344 012737 036624 024504 MOV #EXPHCRC,PSETXT ;...AND HAS A BAD HEADER CRC.
2835 024352 000446 BR 2$
2836
2837 024354 112737 000064 036532 PS4: MOVB #4,TSECX ; TLM SECTOR 4
2838 024362 012737 125252 003440 MOV #<525_7.>!52,INIDA ;...IS CYL 525, SECT 52...
2839 024370 000432 BR 1$ ;...AND IS ERROR FREE.
2840
2841 024372 112737 000065 036532 PS5: MOVB #5,TSECX ; TLM SECTOR 5...

```


CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 N 8
35 -- WRITE CHECK. PAGE 44-4

SEQ 0104

```
2842 024400 012737 135252 003440      MOV    #<565_7.>!52,INIDA ;...IS CYL 565, SECT 52...
2843 024406 000423                BR      1$                ;...AND IS ERROR FREE.
2844
2845 024410                PS7:   PRINTF #TURKY      ; THERE ISN'T ANY SECTOR 7...
(7) 024410 012746 024510                MOV    #TURKY, -(SP)
(6) 024414 012746 000001                MOV    #1, -(SP)
(3) 024420 010600                MOV    SP, RO
(4) 024422 104417                TRAP   CSPNTF
(4) 024424 062706 000004                ADD    #4, SP
2846 024430 012737 000006 003314      MOV    #6, SNGLSEC      ;...CHANGE IT TO 6...
2847 024436 162700 000002      SUB    #2, RO           ;...ADJUST INDEX AND FALL THRU.
2848
2849 024442 112737 000066 036532  PS6:   MOVB   #6, TSECX      ; TLM SECTOR 6...
2850 024450 012737 155555 003440      MOV    #<666_7.>!155,INIDA ;...IS CYL 666, SECT 55, HEAD 1...
2851 024456 005037 024506      1$:    CLR    PSERR
2852 024462 012737 036670 024504      MOV    #EXPNON, PSETXT
2853 024470 016037 024572 024570  2$:    MOV    DTABL(RO), DHEAD ; SET HCRC TO SEARCH FOR.
2854 024476 004737 025174      JSR    PC, FINDSEC     ; GO FIND AND READ IT TO BUF1.
2855 024502 000207                RTS    PC              ; RETURN, 'V' = 1 IF SECTOR NOT FOUND.
2856                .DSABL   LSB
2857
2858 024504 036670                PSETXT: EXPNON        ; EXPECTED ERROR STATUS FOR EACH...
2859 024506 000000                PSERR:  0              ;...IS RETURNED IN THESE LOCATIONS.
2860
2861 024510 047045 040445 042523  TURKY: .ASCIZ  /%NXASECTOR 7 DOESN'T EXIST, USING 6 INSTEAD%N/
2862                .EVEN
```

2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919

024566 025324
024570 006051
024572 006051
024574 044051
024576 132051
024600 174001
024602 036400
024604 113554

024606 005037 003306
024612 005000
024614 077001
024616 104005
024620 113702 025701
024624 004737 025476
024630 025561
024632 004737 025432
024636 113700 004364
024642 001407
024644 120027 000060
024650 103763
024652 120027 000067
024656 101360
024660 110002
024662 110201
024664 013705 003320
024670 042701 177770
024674 001403
024676 062705 000010
024702 077103

024704 110237 025701
024710 012737 024726 000004
024716 012715 000001
024722 000240
024724 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          : ...SELECTED FROM THIS TABLE.
         044051          : 0C29
         132051          : 4829
         174001          : B429
         036400          : F801
         113554          : 3D00
         113554          : 976C

ROMDUMP: CLR PDSW          : CLEAR DUMP REQUEST.
          CLR RO
          SOB RO,,
          EMT+5           : DELAY...
          MOV#5           : ...AND CLEAR KBD.
          JSR HUNIT,R2    : GET DEFAULT UNIT.
          JSR PC,KBDOUT  : SAY "HELLO".
          QO
          JSR PC,KBDIN   : GET A UNIT NUMBER.
          MOV#0           : IF NONE, USE DEFAULT (LAST).
          BEQ 2$
          CMPB RO,#'0     : BR IF INVALID UNIT.
          BLO 1$
          CMPB RO,#'7
          BHI 1$
          MOV#0           : DITTO
          MOV#0           : SAVE NEW UNIT FOR A MOMENT.
          MOV#0 R2,R2
          MOV#0 R2,R1
          MOV TC50,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$

          MOV#0 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.
    
```



```

2920 024726 022626
2921 024730 113737 025701 025536 4$: CMP (SP)+,(SP)+
2922 024732 113737 025701 025536 MOVB HUNIT,NOANS+6
2923 024736 004737 025476 JSR PC,KBDOUT ; SELECTED UNIT DOESN'T ANSWER.
2924 024742 025530 NOANS
2925 024744 000720 BR ROMDUMP
2926 024746 004737 025476 5$: JSR PC,KBDOUT ; GET A SECTOR NUMBER.
2927 024752 025612 Q1
2928 024754 004737 025432 JSR PC,KBDIN
2929 024760 113700 004364 MOVB BUF1,R0
2930 024764 001416 BEQ 6$ ; IF NONE, REDO LAST SECTOR.
2931 024766 120027 000060 CMPB R0,#'0
2932 024772 103765 BLO 5$
2933 024774 120027 000067 CMPB R0,#'7
2934 025000 101362 BHI 5$
2935 025002 110037 025714 MOVB R0,HSECT ; SET IN HEADER STRING.
2936 025006 042700 177770 BIC #'C7,R0
2937 025012 006300 ASL R0 ; SHIFT UP TO WORD OFFSET...
2938 025014 016037 024570 024570 MOV DHEAD(R0),DHEAD ;...AND SET HEADER TO SEARCH FOR.
2939
2940 025022 004737 025476 6$: JSR PC,KBDOUT ; GET AN OUTPUT MODE.
2941 025026 025637 Q2
2942 025030 004737 025432 JSR PC,KBDIN
2943 025034 113700 004364 MOVB BUF1,R0
2944 025040 001411 BEQ 7$ ; IF NONE, USE CURRENT FORMAT.
2945 025042 012737 025324 024566 MOV #OCTMOD,DMODE ; ASSUME OCTAL...
2946 025050 120027 000131 CMPB R0,#'Y
2947 025054 001003 BNE 7$
2948 025056 012737 025350 024566 MOV #HEXMOD,DMODE ;...CHANGE TO HEX IF REQUIRED.
2949 025064
2950 025066 016537 000006 003424 7$: DOIT: MOV 6(R5),TMPO ; GET PROM ID.
2951 025072 004737 025476 JSR PC,KBDOUT ; SHOW UNIT AND SECTOR...
2952 025076 025670 HEADR
2953 025100 012701 003424 MOV #TMPO,R1 ;...AND PROM ID...
2954 025104 004777 177456 JSR PC,ADMODE ;...IN THE CURRENT CONTEXT.
2955 025110 104007 EMT+7 ; <CRLF>
2956 025112 004737 025174 JSR PC,FINDSEC ; SEARCH FOR AND READ SECTOR.
2957 025116 102004 BVC 1$ ; BR IF SECTOR FOUND.
2958 025120 004737 025476 JSR PC,KBDOUT
2959 025124 025731 CANTFIND
2960 025126 000627 BR ROMDUMP
2961
2962 025130 012701 004364 1$: MOV #BUF1,R1 ; NOW FOUND SECTOR IS IN BUF1.
2963 025134 012702 000240 MOV #160.,R2 ; DUMP 160. WORDS AT 8 PER LINE.
2964 025140 104007 EMT+7 ; <CRLF>
2965 025142 004737 025476 2$: JSR PC,KBDOUT
2966 025146 025756 TAB
2967 025150 004777 177412 JSR PC,ADMODE ; PRINT FROM (R1) IN OCTAL OR HEX.
2968 025154 005302 DEC R2
2969 025156 032702 000007 BIT #7,R2
2970 025162 001367 BNE 2$ ; LOOP 'TIL LINE DONE...
2971 025164 104007 EMT+7 ; <CRLF>
2972 025166 005702 TST R2
2973 025170 001364 BNE 2$ ;...AND LOOP 'TIL ALL DONE.
2974 025172 000605 BR ROMDUMP
2975

```



```

2976
2977
2978
2979 025174 012715 000001
2980 025200 012715 000034
2981 025204 005046
2982 025206 005000
2983 025210 012701 006364
2984 025214 020127 005364
2985 025220 001002
2986 025222 012701 006364
2987 025226 005265 000006
2988 025232 016541 000006
2989 025234 021137 024570
2990 025242 001003
2991 025244 005216
2992 025246 012700 000024
2993 025252 077020
2994 025254 012715 000001
2995 025260 012715 000200
2996 025264 005726
2997 025266 001002
2998 025270 000262
2999 025272 000207
3000
3001 025274 012702 004364
3002 025300 012703 000240
3003 025304 020127 006364
3004 025310 001002
3005 025312 012701 005364
3006 025316 012122
3007 025320 077307
3008 025322 000207
3009
3010
3011
3012 025324
(8) 025324 012146
(7) 025326 012746 025770
(6) 025332 012746 000002
(3) 025336 010600
(4) 025340 104417
(4) 025342 062706 000006
3013 025346 000207
3014
3015 025350 010546
3016 025352 012700 025763
3017 025356 012104
3018 025360 005005
3019 025362 012703 000004
3020 025366 006304
3021 025370 006105
3022 025372 077303
3023 025374 062705 000060
3024 025400 120527 000071
3025 025404 003402

```

```

: SEARCH FOR SELECTED SECTOR AND LOAD 160, WORD BLOCK TO BUF1.
: IF UNSUCCESSFUL, RETURN TO CALLER WITH 'V' BIT SET.

```

```

FINDSEC: MOV #1,(R5) ; RESET TLM...
MOV #34,(R5) ; ...AND SET PROM READ MODE.
CLR -(SP) ; CLEAR 'FOUND' FLAG.
CLR R0 ; SET LOOP CONTROL...
MOV #BUFEND,R1 ; NOW, FILL BUF2 FROM TOP DOWN.
1$: CMP R1,#BUF2 ; IN LOOP -- RESET RING POINTER...
BNE 2$ ; ...WHEN IT REACHES THE BOTTOM.
MOV #BUFEND,R1
2$: INC 6(R5) ; INCR ROM ADDRESS...
MOV 6(R5),-(R1) ; ...AND GET A WORD.
CMP (R1),DHEAD ; IS THIS THE DESIRED HCRC WORD ??
BNE 3$ ; BR IF NOT.
INC (SP) ; YES, SET 'FOUND' FLAG...
MOV #20,R0 ; ...AND SET TO READ 19 MORE.
3$: SOB R0,1$ ; LOOP 'TIL DONE...
MOV #1,(R5) ; ...THEN, RESET TLM...
MOV #200,(R5) ; ...AND RESTORE DRIVE READY.
TST (SP)+ ; SECTOR FOUND ??
BNE 4$ ; BR IF SO.
SEV ; NO, SET 'V' BIT...
RTS PC ; ...AND RETURN.

4$: MOV #BUF1,R2 ; NOW COPY FOUND SECTOR TO BUF1.
MOV #160,R3
5$: CMP R1,#BUFEND ; IN LOOP -- RESET RING POINTER...
BNE 6$ ; ...WHEN IT REACHES THE TOP.
MOV #BUF2,R1
6$: MOV (R1)+,(R2)+ ; BUF2 => BUF1.
SOB R3,5$ ; RETURN, 'V' IS CLEAR.
RTS PC

```

```

: SUBROUTINES TO DUMP DATA IN OCTAL OR HEX.

```

```

OCTMOD: PRINTF #OCTTXT,(R1)+

```

```

MOV (R1)+,-(SP)
MOV #OCTTXT,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP

```

```

RTS PC
HEXMOD: MOV R5,-(SP) ; SAVE TLM POINTER.
1$: MOV #HEXTXT,R0 ; SET HEX TEXT POINTER.
MOV (R1)+,R4 ; NEXT WORD => R4.
2$: CLR R5 ; ...AND PARSE EACH NIBBLE IN R5.
MOV #4,R3
3$: ASL R4
ROL R5 ; NIBBLE => R5<3:0>
SOB R3,3$
ADD #10,R5 ; CONVERT TO ASCII 0 TO 9...
CMPB R5,#'9'
BLE 4$

```

3026 025406 062705 000007
3027 025412 110520
3028 025414 105710
3029 025416 001360
3030 025420 004737 025476
3031 025424 025763
3032 025426 012605
3033 025430 000207
3034 025432 012701 004364
3035 025436 104005
3036 025440 110011
3037 025442 001775
3038 025444 120027 000003
3039 025446 001411
3040 025450 120027 000015
3041 025452 001403
3042 025456 104004
3043 025460 005201
3044 025462 000764
3045 025464 105011
3046 025466 104007
3047 025470 000207
3048 025472 104444
3049 025474 104444
3050 (3)
3051 025476 017646 000000
3052 025476 012746 025774
3053 (8) 025502 012746 000002
3054 (7) 025506 010600
3055 (6) 025512 104417
3056 (3) 025514 062706 000006
3057 (4) 025516 062716 000002
3058 (4) 025522 000207
3059 025526
3060 025530 020040 046124 035115
3061 025561 015 012
3062 025563 040 052040 046514
3063 025612 020040 042523 052103
3064 025637 040 044040 054105
3065 025670 005015 052411 044516
3066 025701 060 020040 042523
3067 025714 020061 050040 047522
3068 025731 011 040503 023516
3069 025756 000011
025760 005015 000
025763 101 041502 000104
025770 047445 000066
025774 052045 000
026000

```
4$: ADD #7,R5 ;...AND A(10) TO F(15)...  
MOV R5,(R0)+ ;...AND STUFF IT.  
TSTB (R0)  
BNE 2$ ; LOOP 'TIL 4 CHARS ASSEMBLED...  
JSR PC,KBDOUT ;...THEN DUMP IT.  
HEXTXT  
MOV (SP)+,R5 ; RESTORE R5.  
RTS PC  
  
KBDIN: MOV #BUF1,R1 ; TEMP ASCII BUFFER.  
1$: EMT+5 ; CHAR => R0.  
MOV R0,(R1)  
BEQ 1$  
CMPB R0,#3  
BEQ 3$ ; EXIT ON <^C>.  
CMPB R0,#15  
BEQ 2$ ; BR ON TERMINATOR.  
EMT+4 ; OTHERWISE, ECHO...  
INC R1 ;...BUMP POINTER...  
BR 1$ ; AND LOOP.  
2$: CLRB (R1) ; TERMINATE INPUT STRING...  
EMT+7 ;...ECHO CRLF...  
RTS PC ;...AND RETURN TO CALLER.  
3$: DOCLN ; EXIT ON <^C>.
```

KBDOUT: PRINTF #PURE,a(SP)

```
TRAP CSDCLN  
MOV a(SP),-(SP)  
MOV #PURE,-(SP)  
MOV #2,-(SP)  
MOV SP,R0  
TRAP C$PNTF  
ADD #6,SP
```

```
ADD #2,(SP)  
RTS PC  
  
NOANS: .ASCIZ / TLM:0 DOESN'T ANSWER/<15><12>  
Q0: .ASCII <15><12>  
Q1: .ASCIZ / TLM UNIT (0 TO 7) ? /  
Q2: .ASCIZ / SECTOR (1 TO 6) ? /  
HEADR: .ASCIZ / HEX OUTPUT (Y OR N) ? /  
HUNIT: .ASCII <15><12><11>/UNIT: /  
HSECT: .ASCII /0 SECTOR: /  
CANTFIND: .ASCIZ /1 PROM ID: /  
TAB: .ASCIZ <11>  
CRLF: .ASCIZ <15><12>  
HEXTXT: .ASCIZ /ABCD/  
OCTTXT: .ASCIZ /806/  
PURE: .ASCIZ /8T/  
.EVEN
```

3071			
3072			
3073			
3074			
3075			
3076	026000	013700	003402
3077	026004	001414	
3078	026006	020027	000002
3079	026012	103406	
3080	026014	005737	003344
3081	026020	001403	
3082	026022	012700	000043
3083	026026	000402	
3084	026030	012700	000035
3085	026034	000402	
3086	026036	012700	000026
3087	026042	010037	026050
3088	026046	000207	
3089			
3090	026050		
(4)	026050	000043	
(3)	026052		
(6)	026052	012114	
(6)	026054	012200	
(6)	026056	012264	
(6)	026060	012350	
(6)	026062	012434	
(6)	026064	012556	
(6)	026066	013020	
(6)	026070	013140	
(6)	026072	013244	
(6)	026074	013332	
(6)	026076	013436	
(6)	026100	013664	
(6)	026102	014050	
(6)	026104	014216	
(6)	026106	014410	
(6)	026110	014612	
(6)	026112	015036	
(6)	026114	015262	
(6)	026116	015552	
(6)	026120	015746	
(6)	026122	016160	
(6)	026124	016246	
(6)	026126	016414	
(6)	026130	016550	
(6)	026132	016634	
(6)	026134	016762	
(6)	026136	017662	
(6)	026140	020306	
(6)	026142	020502	
(6)	026144	021276	
(6)	026146	021650	
(6)	026150	022054	
(6)	026152	022222	
(6)	026154	022502	

.SBTTL
 .SBTTL SUPERVISOR DISPATCH TABLE.

... SUBROUTINE TO ADJUST THE NUMBER OF TESTS TO RUN IAW CONTROLLER TYPE.

```

ADJTN:  MOV    RLTP,R0      : GET RL TYPE.
        BEQ    2$          : BR IF RL11.
        CMP    R0,#RLV12   :
        BLO    1$          : BR IF RLV11.
        TST    TCSR        :
        BEQ    1$          : BR IF TLM BYPASSED.
        MOV    #35.,R0     : RLV12 WITH TLM RUNS ALL.
1$:     SKP2
        MOV    #29.,R0     : RLV12 OR RLV11 RUN THRU 29.
        SKP2
2$:     MOV    #22.,R0     : RL11 RUN THRU 22.
        MOV    R0,L$DISPATCH-2 : ...AND SET LAST TEST NUMBER.
        RTS    PC
  
```

DISPATCH T\$TESTNUM

L\$DISPATCH::	WORD	35
	.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

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07⁶ ⁹ PAGE 45-1
SUPERVISOR DISPATCH TABLE.

SEQ 0110

(6) 026156 023364

.WORD T35

```
3092      .SBTTL GLOBAL ERROR HANDLERS AND ASCII TEXT
3093      :
3094      : THESE ARE THE HANDLERS FOR THE VARIOUS ERROR SIGNATURES.
3095      :
3096      BGNMSG ERRSIG
3097      (3) 026160 000137 027306      NOSIG: JMP CKERLT      ; SOME HAVE NO SIGNATURE. ERRSIG::
3098      (3) 026160 000137 027100      ERRO:  JMP ALLREGS    ; ALL REGISTERS.
3099      (3) 026164 000137 027100      ERR1:  PRINTB #FRMT2,GDDAT ; BUS-TIMEOUT REG ADDR.
3100      (8) 026170 013746 003432      MOV      GDDAT,-(SP)
3101      (7) 026174 012746 027501      MOV      #FRMT2,-(SP)
3102      (6) 026200 012746 000002      MOV      #2,-(SP)
3103      (3) 026204 010600      MOV      SP,R0
3104      (4) 026206 104414      TRAP    C$PNTB
3105      (4) 026210 062706 000006      ADD     #6,SP
3106      (4) 026214 000137 027306      JMP     CKERLT
3107      (9) 026220 013746 003434      ERR2:  PRINTB #FRMT4,GDDAT,BDDAT ; EXP'D VS REC'D.
3108      (8) 026224 013746 003432      MOV      BDDAT,-(SP)
3109      (7) 026230 012746 027602      MOV      GDDAT,-(SP)
3110      (6) 026234 012746 000003      MOV      #FRMT4,-(SP)
3111      (3) 026240 010600      MOV      #3,-(SP)
3112      (4) 026242 104414      MOV      SP,R0
3113      (4) 026244 062706 000010      TRAP    C$PNTB
3114      (4) 026244 062706 000010      ADD     #10,SP
3115      (4) 026250 000137 027306      JMP     CKERLT
3116      (8) 026254 013746 003434      ERR3:  PRINTB #FRMT5,BDDAT ; MAINT SEQ FAILED STATE.
3117      (7) 026260 012746 027636      MOV      BDDAT,-(SP)
3118      (6) 026264 012746 000002      MOV      #FRMT5,-(SP)
3119      (3) 026270 010600      MOV      #2,-(SP)
3120      (4) 026272 104414      MOV      SP,R0
3121      (4) 026274 062706 000006      TRAP    C$PNTB
3122      (4) 026274 062706 000006      ADD     #6,SP
3123      (3) 026300 000137 027100      JMP     ALLREGS
3124      (9) 026304 013746 003434      ERR4:  PRINTB #FRMT4,GDDAT,BDDAT ; EXP'D VS REC'D...
3125      (8) 026310 013746 003432      MOV      BDDAT,-(SP)
3126      (7) 026314 012746 027602      MOV      GDDAT,-(SP)
3127      (6) 026320 012746 000003      MOV      #FRMT4,-(SP)
3128      (3) 026324 010600      MOV      #3,-(SP)
3129      (4) 026326 104414      MOV      SP,R0
3130      (4) 026330 062706 000010      TRAP    C$PNTB
3131      (4) 026330 062706 000010      ADD     #10,SP
3132      (3) 026334 000137 027100      JMP     ALLREGS ;...AND REGISTERS.
3133      (6) 026340 012746 034404      ERR6:  PRINTB #FRMT3,#_199 ; CS ERRORS.
3134      (7) 026344 012746 027517      MOV      #EM99,-(SP)
3135      (6) 026350 012746 000002      MOV      #FRMT3,-(SP)
3136      (3) 026354 010600      MOV      #2,-(SP)
3137      (4) 026356 104414      MOV      SP,R0
3138      (4) 026360 062706 000006      TRAP    C$PNTB
3139      (4) 026364 000137 027100      ADD     #6,SP
3140      (4) 026364 000137 027100      JMP     ALLREGS
```

3115
3116 026370
(8) 026370 013746 003424
(7) 026374 012746 027670
(6) 026400 012746 000002
(3) 026404 010600
(4) 026406 104414
(4) 026410 062706 000006
3117 026414 000137 027306
3118
3119 026420
(8) 026420 012746 034404
(7) 026424 012746 027541
(6) 026430 012746 000002
(3) 026434 010600
(4) 026436 104414
(4) 026440 062706 000006
3120 026444 000137 027100
3121
3122 026450
(8) 026450 013746 003434
(7) 026454 012746 027723
(6) 026460 012746 000002
(3) 026464 010600
(4) 026466 104414
(4) 026470 062706 000006
3123 026474 000137 027306
3124
3125 026500 004737 026610
3126 026504
(8) 026504 012746 034404
(7) 026510 012746 030241
(6) 026514 012746 000002
(3) 026520 010600
(4) 026522 104414
(4) 026524 062706 000006
3127 026530 000137 027100
3128
3129 026534 004737 026610
3130 026540 000137 026340
3131
3132 026544 004737 026610
3133 026550
(10) 026550 013746 003434
(9) 026554 013746 003432
(8) 026560 013746 003424
(7) 026564 012746 030276
(6) 026570 012746 000004
(3) 026574 010600
(4) 026576 104414
(4) 026600 062706 000012
3134 026604 000137 027100
3135
3136
3137
3138 026610 005737 003430

ERR7: PRINTB #FRMT6,TMPO ; CPU PRIORITY.

MOV TMPO, -(SP)
MOV #FRMT6, -(SP)
MOV #2, -(SP)
MOV SP, RO
TRAP CSPNTB
ADD #6, SP

JMP CKERLT

ERR10: PRINTB #FRMT3A,#EM99 ; EXPECTED OPI ERRORS.

MOV #EM99, -(SP)
MOV #FRMT3A, -(SP)
MOV #2, -(SP)
MOV SP, RO
TRAP CSPNTB
ADD #6, SP

JMP ALLREGS

ERR11: PRINTB #FRMT10,BDDAT ; OPI TIMING ERROR.

MOV BDDAT, -(SP)
MOV #FRMT10, -(SP)
MOV #2, -(SP)
MOV SP, RO
TRAP CSPNTB
ADD #6, SP

JMP CKERLT

ERR12: JSR PC,PRTPA ; STAT WRONG ON NXM (ABOVE 32K)...
PRINTB #FRMT21,#EM99 ; ...OR BBS7/NXM FAILURE.

MOV #EM99, -(SP)
MOV #FRMT21, -(SP)
MOV #2, -(SP)
MOV SP, RO
TRAP CSPNTB
ADD #6, SP

JMP ALLREGS

ERR13: JSR PC,PRTPA ; UNEX STAT ERRORS (ABOVE 32K).
JMP ERR6

ERR14: JSR PC,PRTPA ; DATA WRONG (ABOVE 32K).
PRINTB #FRMT22,TMPO,GDDAT,BDDAT ; EXPD VS RECD.

MOV BDDAT, -(SP)
MOV GDDAT, -(SP)
MOV TMPO, -(SP)
MOV #FRMT22, -(SP)
MOV #4, -(SP)
MOV SP, RO
TRAP CSPNTB
ADD #12, SP

JMP ALLREGS

... PACK AND PRINT 22 BIT PHYSICAL ADDRESS FROM TMP1..TMP2.

PRTPA: TST TMP2 ; BIT 15 ON ??

3139 026614 100003
3140 026616 062737 100000
3141 026624 006137 003426
3142 026630
(9) 026630 013746 003430
(8) 026634 013746 003426
(7) 026640 012746 030201
(6) 026644 012746 000003
(3) 026650 010600
(4) 026652 104414
(4) 026654 062706 000010
3143 026660 000207
3144
3145
3146
3147 026662 004737 027042
3148 026666
3149 026666
(9) 026666 012746 034404
(8) 026672 013746 003426
(7) 026676 012746 030410
(6) 026702 012746 000003
(3) 026706 010600
(4) 026710 104415
(4) 026712 062706 000010
3150 026716
(7) 026716 012746 030431
(6) 026722 012746 000001
(3) 026726 010600
(4) 026730 104415
(4) 026732 062706 000004
3151 026736 000137 027100
3152
3153 026742 004737 027042
3154 026746 000137 027306
3155
3156 026752 004737 027042
3157 026756
(11) 026756 013746 003434
(10) 026762 013746 003432
(9) 026766 013746 003424
(8) 026772 013746 003426
(7) 026776 012746 030031
(6) 027002 012746 000005
(3) 027006 010600
(4) 027010 104415
(4) 027012 062706 000014
3158 027016
(7) 027016 012746 030431
(6) 027022 012746 000001
(3) 027026 010600
(4) 027030 104415
(4) 027032 062706 000004
3159 027036 000137 027100
3160
3161 027042

003430 1\$: BPL 1\$: NO 'C' = 0;
ADD #BIT15,TMP2 : YES 'C' = 1;
ROL TMP1 : TMP1 = ADDR<21:15>;
PRINTB #FRMT20,TMP1,TMP2 ; TMP2 = ADDR<14:00>.

RTS PC

... TLM UNIQUE ERROR SIGNATURES.

ERR20: JSR PC,E2022 : BAD STATUS AFTER READ,READNH,WRTCHK...
ERR20A: PRINTX #FRMT26,TMP1,#EM99 : ...OR AFTER SEEK,GET STATUS,WRITE.

PRINTX #FRMT27 : CRLF.

JMP ALLREGS

ERR21: JSR PC,E2022 : READ HEADER FAILS.
JMP CKERLT

ERR22: JSR PC,E2022 : DATA FROM TLM WRONG.
PRINTX #FRMT14,TMP1,TMP0,GDDAT,BDDAT

PRINTX #FRMT27 : CRLF

JMP ALLREGS

E2022: PRINTX #FRMT25,#TSEC,TMP2,#TPEAK

MOV TMP2,-(SP)
MOV TMP1,-(SP)
MOV #FRMT20,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTB
ADD #10,SP

MOV #EM99,-(SP)
MOV TMP1,-(SP)
MOV #FRMT26,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #10,SP

MOV #FRMT27,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #4,SP

MOV BDDAT,-(SP)
MOV GDDAT,-(SP)
MOV TMP0,-(SP)
MOV TMP1,-(SP)
MOV #FRMT14,-(SP)
MOV #5,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #14,SP

MOV #FRMT27,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C\$PNTX
ADD #4,SP

(10) 027042 012746 036602
(9) 027046 013746 003430
(8) 027052 012746 036523
(7) 027056 012746 030401
(6) 027062 012746 000004
(3) 027066 010600
(4) 027070 104415
(4) 027072 062706 000012
3162 027076 000207
3163
3164
3165
3166
3167 027100 005000
3168 027102 023727 003402 000003
3169 027110 001002
3170 027112 012700 000045
3171 027116 110037 027425
3172 027122 110037 027467
3173 027126
(7) 027126 012746 027366
(6) 027132 012746 000001
(3) 027136 010600
(4) 027140 104414
(4) 027142 062706 000004
3174 027146
(13) 027146 013746 003360
(12) 027152 013746 003356
(11) 027156 013746 003354
(10) 027162 013746 003352
(9) 027166 013746 003350
(8) 027172 012746 030434
(7) 027176 012746 027436
(6) 027202 012746 000007
(3) 027206 010600
(4) 027210 104414
(4) 027212 062706 000020
3175 027216
(13) 027216 013746 003376
(12) 027222 013746 003372
(11) 027226 013746 003370
(10) 027232 013746 003366
(9) 027236 013746 003364
(8) 027242 012746 030455
(7) 027246 012746 027436
(6) 027252 012746 000007
(3) 027256 010600
(4) 027260 104414
(4) 027262 062706 000020
3176 027266
(7) 027266 012746 027476
(6) 027272 012746 000001
(3) 027276 010600
(4) 027300 104414
(4) 027302 062706 000004
3177

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 #1,R0 ;...OR ENABLE THE EXTRA REGISTER.
1\$: MOVB R0,FRMT1A ; ENABLE/DISABLE BAE TEXT.
MOVB R0,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

```
3178  
3179  
3180  
3181 027306 104420  
3182 027310 103425  
3183 027310 103425 003304  
3184 027312 005737  
3185 027316 001422  
3186 027320 005277 154142  
3187 027324 027737 154136 003304  
3188 027332 002414  
3189 027334 012746 030350  
3190 027340 012746 000001  
3191 027344 010600  
3192 027346 104417  
3193 027350 062706 000004  
3194 027354 013700 003322  
3195 027360 104451  
3196 027362 104444  
3197 027364 104444  
3198 027364 104423
```

...AND CHECK ERROR LIMIT (IF ANY).
KERLT: INLOOP
BCOMplete 3\$: RETURN IF LOOPING... TRAP CSINLP
TST ERRLMT :...OR NO ERROR LIMIT SET.
BEQ 3\$:COUNT THE UNIT ERROR DETECTED
INC @ERPOINT :REACHED THE ERROR LIMIT?
CMP @ERPOINT,ERRLMT :NO, RETURN
BLT 3\$: "ERROR LIMIT EXCEEDED"
PRINTF #FRMT23
MOV #FRMT23, -(SP)
MOV #1, -(SP)
MOV SP, R0
TRAP CSPTF
ADD #4, SP
2\$: DODU UNITST :DROP THE UNIT... MOV UNITST, R0
TRAP CSDODU
DOCLN :...AND ABORT. TRAP CSDCLN
3\$: ENDMSG : OR RETURN TO CALLER. L10063: TRAP CMSG

3193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
3248

027366 040445 042522 044507
027425 045 004501 020040
027436 047045 052045 047445
027467 045 031123 047445
027476 047045 000
027501 045 040501 042104
027517 045 042501 051122
027541 045 042501 050130
027602 040445 054105 023520
027636 040445 052101 044440
027670 040445 052101 050040
027723 045 047501 044520
027766 040445 054105 042520
030031 045 022516 053501
030120 047045 042045 022463
030201 045 040501 020124
030241 045 042501 050130
030276 040445 047527 042122
030350 040445 051105 047522

030401 045 022524 022524
030410 047045 052045 040445
030431 045 000116

... FORMATTED ASCII TEXT.

FRMT1: .ASCII /%REGISTERS CS BA DA MP/
FRMT1A: .ASCII /% BAE/
FRMT1B: .ASCII /%NXT%06%S2%06%S2%06%S2%06%/
FRMT1C: .ASCII /%S2%06%/
FRMT1D: .ASCII /%N/
FRMT2: .ASCII /%AADDR: %06%N/
FRMT3: .ASCII /%AERRORS SET: %T%N/
FRMT3A: .ASCII /%AEXP'D: ERR HNF OPI REC'D: %T%N/
FRMT4: .ASCII /%AEXP'D: %06% REC'D: %06%N/
FRMT5: .ASCII /%AAT INTERNAL STATE %D2%N/
FRMT6: .ASCII /%AAT PROCESSOR LEVEL %01%N/
FRMT10: .ASCII /%AOPI FLAG RECEIVED AT %D4% MSEC. %N/
.ASCII /%AEXPECTED FROM 155 TO 650 MSEC. %N/
FRMT14: .ASCII /%N%WORD: %03% ADDR: %06% EXP'D: %06% REC'D: %06%
FRMT15: .ASCII /%N%D3% WORDS BAD OUT OF 255 WORDS TRANSFERRED %N/
FRMT20: .ASCII /%AAT PHYSICAL ADDRESS: %03%05%N/
FRMT21: .ASCII /%AEXP'D: ERR NXM REC'D: %T%N/
FRMT22: .ASCII /%AWORD: %03% EXP'D: %06% REC'D: %06%N/
FRMT23: .ASCII /%AERROR LIMIT EXCEEDED %N/

FRMT25: .ASCII /%T%T%T%/
FRMT26: .ASCII /%NXT% REC'D: %T%/
FRMT27: .ASCII /%N/
.EVEN

... PURE (UNFORMATTED) ASCII TEXT.

BREG: .ASCII /BEFORE COMMAND: /
AFREG: .ASCII /AFTER COMMAND: /
CRTIM: .ASCII /CONTROLLER TIMED OUT/
DRTIM: .ASCII /DRIVE READY TIMED OUT/
CERR: .ASCII / ERR/
DEMES: .ASCII / DRV/
NXMMES: .ASCII / NXM/
PARMES: .ASCII / PAR/
DLTMES: .ASCII / DLT/
HNFMES: .ASCII / HNF/
DCRCMES: .ASCII / DCRC/
HCRCMES: .ASCII / HCRC/
OPIMES: .ASCII / OPI/
NON: .ASCII / NONE/

EM0: .ASCII /CONTROLLER DOESN'T ANSWER -- CAN'T TEST/
EM1: .ASCII /CAN NOT ADDRESS RLCS/
EM2: .ASCII /CAN NOT ADDRESS RLBA/
EM3: .ASCII /CAN NOT ADDRESS RLDA/
EM4: .ASCII /CAN NOT ADDRESS RLMP/
EM4A: .ASCII /CAN NOT ADDRESS RLBAE/
EM4B: .ASCII /BAE SHOULD BE DISABLED -- IT'S NOT !!!/
EM5: .ASCII \RLCS READ/WRITE ERROR (BIT 0 DON'T CARE)\
EM6: .ASCII \RLBA READ/WRITE ERROR\
EM7: .ASCII \RLDA READ/WRITE ERROR\
EM8: .ASCII \RLBAE READ/WRITE ERROR\

3249	031303	122	041114	020101	EM10:	.ASCIZ	/RLBA ERROR AFTER MAINT. FUNCTION/
3250	000000	046123	040504	042440	EM12:	.ASCIZ	/RLDA ERROR AFTER MAINT. FUNCTION/
3251	000000	122	041514	020123	EM14:	.ASCIZ	/RLCS INCORRECT AFTER NO-OP/
3252	000000	1440	047516	020120	EM14A:	.ASCIZ	/NO-OP ALTERED RLBA, RLDA, OR RLMP/
3253	000000	1502	040515	042524	EM14B:	.ASCIZ	/MAINTENANCE MODE SEQUENCER FAILED/
3254	000000	1544	047516	052116	EM15:	.ASCIZ	/NO INTERRUPT ON FUNCTION COMPLETE/
3255	000000	1606	047111	051122	EM16:	.ASCIZ	/INTERRUPT PRIORITY FAILURE/
3256	000000	1641	051117	042503	EM17:	.ASCIZ	/FORCED OPI -- INTERRUPT NOT RECEIVED/
3257	000000	1706	020103	043117	EM20:	.ASCIZ	/CRC OF DA+3 INCORRECT (SERIAL DATA PATH)/
3258	000000	1757	041522	047440	EM21:	.ASCIZ	/CRC OF CRC OF DA+4 INCORRECT (SERIAL DATA PATH)/
3259	000000	2037	106	027514	EM22:	.ASCIZ	\FILL/EMPTY FIFO DATA TRANSFER ERROR\
3260	000000	2103	106	027514	EM23:	.ASCIZ	\FILL/EMPTY FIFO LAST WORD+1 INCORRECT\
3261	000000	2151	105	051117	EM27:	.ASCII	/ERROR STATUS INCORRECT AFTER FORCED OPI/<15><12>
3262	000000	2222	047111	040511		.ASCII	/INITIAL WORD COUNT WAS /
3263	000000	2251	074	030461	EM27X:	.ASCIZ	/< 511.7
3264	000000	2360	050117	020111	EM31:	.ASCIZ	/OPI FLAG NOT RECEIVED/
3265	000000	2386	050117	044524	EM32:	.ASCIZ	/OPI TIMING INCORRECT/
3266	000000	2386	127	044524	EM44:	.ASCIZ	/WRITING RLMP MODIFIED RLCS/
3267	000000	2386	051127	047111	EM45:	.ASCIZ	/WRITING RLMP MODIFIED RLBA/
3268	000000	2421	127	044524	EM46:	.ASCIZ	/WRITING RLMP MODIFIED RLDA/
3269	000000	2451	051127	047111	EM47:	.ASCIZ	/WRITING RLMP MODIFIED RLBAE/
3270	000000	2510	046122	053440	EM50:	.ASCIZ	/RLCS WRONG AFTER WRITING RLBAE/
3271	000000	2547	122	042501	EM51:	.ASCIZ	/RLBAE MODIFIED RLBA/
3272	000000	2573	122	042501	EM52:	.ASCIZ	/RLBAE MODIFIED RLDA/
3273	000000	2617	042	020124	EM61:	.ASCIZ	/"BIT SET" ON RLCS YIELDS WRONG RESULT/
3274	000000	2655	042	020124	EM62:	.ASCIZ	/"BIT CLEAR" ON RLCS YIELDS WRONG RESULT/
3275	000000	2735	042	020124	EM63:	.ASCIZ	/"BIT SET" ON RLBA YIELDS WRONG RESULT/
3276	000000	2765	042	020124	EM64:	.ASCIZ	/"BIT CLEAR" ON RLBA YIELDS WRONG RESULT/
3277	000000	2805	042	020124	EM65:	.ASCIZ	/"BIT SET" ON RLDA YIELDS WRONG RESULT/
3278	000000	2821	042	020124	EM66:	.ASCIZ	/"BIT CLEAR" ON RLDA YIELDS WRONG RESULT/
3279	000000	2871	042	020124	EM67:	.ASCIZ	/"BIT SET" ON RLBAE YIELDS WRONG RESULT/
3280	000000	2940	041042	041440	EM68:	.ASCIZ	/"BIT CLEAR" ON RLBAE YIELDS WRONG RESULT/
3281	000000	2981	102	051523	EM69:	.ASCIZ	/BUS-RESET DIDN'T CLEAR RLCS/
3282	000000	2981	102	051523	EM70:	.ASCIZ	/BUS-RESET DIDN'T CLEAR RLBA/
3283	000000	2981	102	051523	EM71:	.ASCIZ	/BUS-RESET DIDN'T CLEAR RLDA/
3284	000000	2981	102	051523	EM71A:	.ASCIZ	/BUS-RESET DIDN'T CLEAR RLBAE/
3285	000000	3045	051127	047111	EM72:	.ASCIZ	/WRITING RLCS MODIFIED RLBA/
3286	000000	3072	127	044524	EM73:	.ASCIZ	/WRITING RLCS MODIFIED RLDA/
3287	000000	3080	046122	020105	EM73A:	.ASCIZ	/RLBAE WRONG AFTER WRITING RLCS/
3288	000000	3080	127	044524	EM74:	.ASCIZ	/WRITING RLBA MODIFIED RLCS/
3289	000000	3080	127	044524	EM75:	.ASCIZ	/WRITING RLBA MODIFIED RLDA/
3290	000000	3080	127	044524	EM75A:	.ASCIZ	/WRITING RLBA MODIFIED RLBAE/
3291	000000	3080	127	044524	EM76:	.ASCIZ	/WRITING RLDA MODIFIED RLCS/
3292	000000	3080	127	044524	EM77:	.ASCIZ	/WRITING RLDA MODIFIED RLBA/
3293	000000	3080	127	044524	EM77A:	.ASCIZ	/WRITING RLDA MODIFIED RLBAE/
3294	000000	3080	116	042440	EM80:	.ASCII	/NXM ERROR FLAG DIDN'T SET/
3295	000000	3080	020054	040502	EM80X:	.ASCIZ	/ BANK SELECT 7 (BBS7) FAILS/
3296	000000	3080	123	040524	EM90:	.ASCIZ	/STATUS INCORRECT AFTER NON-EX MEMORY ACCESS/
3297	000000	3080	123	042516	EM91:	.ASCIZ	/UNEXPECTED STATUS ERROR ON EXT MEMORY ACCESS/
3298	000000	3080	040504	044440	EM92:	.ASCIZ	\DATA INCORRECT TO/FROM EXTENDED MEMORY\
3299	000000	3080	125	050130	EM98:	.ASCIZ	/UNEXPECTED CONTROLLER ERRORS/
3300	034404	000170	042516	050130	EM99:	.BLKB	120.
3301							
3302							
3303							
3304	034574	052502	020123	044524	EM100:	.ASCIZ	/BUS TIME-OUT ON TLM ADDRESS/

... TLM UNIQUE ERROR TEXT.

3305	034630	046124	020115	052123	EM100A:	.ASCII	/TLM STATUS WRONG/<15><12>
3306	034630	044103	041505	020113		.ASCII	/CHECK THAT CABLE IS IN PLACE !!!!/
3307	034630	120	047522	020111	EM100B:	.ASCII	/PROM ID WRONG/<15><12>
3308	034630	044103	041505	020111		.ASCII	/CHECK THAT CORRECT PROM SET IS INSTALLED !!!!/
3309	035007	123	051531	046103	EM101:	.ASCII	\SYSClk<11> AND/OR PWROK<8> NOT SET IN TLM STATUS\
3310	035070	051104	053111	020105	EM102:	.ASCII	/DRIVE READY STATUS BIT<0> INCORRECT IN RLCSR/
3311	035145	104	044522	042526	EM103:	.ASCII	/DRIVE ERROR STATUS BITS<15:14> INCORRECT IN RLCSR/
3312	035225	104	044522	042526	EM104:	.ASCII	/DRIVE
3313	035235	060	051440	046103	EM104X:	.ASCII	/0 SELECT BITS<10:9> INCORRECT IN TLM STATUS/
3314	035235	104	044522	042526	EM105:	.ASCII	/DRIVE COMMAND (GET STATUS) INCORRECT IN TLM CSR+2/
3315	035373	122	052105	051125	EM106:	.ASCII	/RETURNED DRIVE STATUS INCORRECT IN RLMP/
3316	035443	105	051122	051117	EM107:	.ASCII	/ERRORS IN RLCSR AFTER "GET STATUS" COMMAND/
3317	035516	051104	053111	020105	EM110:	.ASCII	/DRIVE COMMAND (SEEK DIFF) INCORRECT IN TLM CSR+2/
3318	035577	105	051122	051117	EM111:	.ASCII	/ERRORS IN RLCSR AFTER "SEEK" COMMAND/
3319	035644	051127	052111	020105	EM112:	.ASCII	/WRITE GATE<13> OR DATA ACTIVE<12> DIDN'T SET DURING WRITE/
3320	035736	051127	052111	020105	EM113:	.ASCII	/WRITE GATE ERROR BIT<14> SET DURING WRITE/
3321	036010	051127	052111	020105	EM113A:	.ASCII	/WRITE GATE BIT<13> STILL SET AFTER WRITE DONE/
3322	036066	051105	047522	051522	EM114:	.ASCII	/ERRORS IN RLCSR AFTER WRITE DATA COMMAND/
3323	036137	105	051122	051117	EM115:	.ASCII	/ERROR STATUS INCORRECT AFTER READ DATA/
3324	036205	000	044527	044124	EM115X:	.ASCII	<0>/WITHOUT HEADER CHECK/
3325	036238	122	040505	020104	EM116:	.ASCII	/READ HEADER CAN'T FIND TLM SECTOR 2 (CYL1,SEC6)/
3326	036313	122	041505	044505	EM117:	.ASCII	/RECEIVED DATA INCORRECT AFTER READ/
3327	036355	000	044527	044124	EM117X:	.ASCII	<0>/WITHOUT HEADER CHECK/
3328	036403	105	051122	051117	EM118:	.ASCII	/ERROR STATUS INCORRECT AFTER WRITE-CHECK/
3329	036454	051120	046517	051040	EM119:	.ASCII	/PROM READ FAILURE/<15><12>
3330	036477	103	047101	052047		.ASCII	/CAN'T FIND SECTOR /
3331	036521	001	000		EM119X:	.ASCII	/1/
3332	036523	123	041505	047524	TSEC:	.ASCII	/SECTOR /
3333	036532	026061	000040		TSECX:	.ASCII	/1 /
3334	036536	047516	044515	040516	TCLK0:	.ASCII	/NOMINAL CLOCK/
3335	036554	040506	052123	041440	TCLK1:	.ASCII	/FAST CLOCK/
3336	036567	123	047514	020127	TCLK2:	.ASCII	/SLOW CLOCK/
3337	036602	026440	0464 0	054101	TPEAK:	.ASCII	/ - MAX PEAK SHIFT/
3338	036624	054105	023520	035104	EXPHCRC:	.ASCII	/EXP'D: ERR HCRC OPI/
3339	036650	054105	023520	035104	EXPDCRC:	.ASCII	/EXP'D: ERR DCRC/
3340	036670	054105	023520	035104	EXPNON:	.ASCII	/EXP'D: NONE/
3341						.EVEN	


```

3343 .SBTTL DEVELOPMENT/DEBUG AIDS
3344 .SBTTL
3345 .SBTTL RLV12 EMULATOR
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393
3394
3395
3396
3397
3398
036704 036704
036704 000000
036706 023727 003402 000000
036714 001004
036716 042777 036000 144400
036724 000517
036726 027727 144400 177001 1$:
036734 001404
036736 052777 112000 144360
036744 000507
036746 020527 020416 2$:
036752 001004
036754 052777 120000 144342
036762 000500
036764 012701 004364 3$:
036770 012702 005364
036774 012703 037236
037000 023701 010352
037004 001411
037006 012701 100000
037012 012702 101000
037016 012737 037106 000004
037024 005237 177572
037030 062777 000002 144270 4$:
037036 005577 144272
037042 012123
037044 000240
037046 020327 040236
037052 103766
037054 012703 037236
037060 062777 000002 144240 14$:
037066 005577 144242
037072 012322
037074 000240
037076 020327 040234
037102 103766
037104 000406
037106 052777 120000 144210 5$:
037114 012716 037122
037120 000002
037122 005037 177572 000004 6$:
037126 012737 010624
037134 017700 144170
037140 062700 000006
037144 110077 144160
PRINT ; FOR DEBUG, MIMIC FLAG IS HERE...
MIMIC: 0 ;...SET NZ TO ENABLE EMULATOR.
EMURLV: CMP RLTP,#RL11
BNE 1$ ; BR IF RLV
BIC #36000,@RLCS ; RL11 -- NOP -- CLEAR ERROR BITS.
BR 7$ ; AND THAT'S ALL.
1$: CMP @RLMP,#-511. ; WORD COUNT RIGHT ??
BEQ 2$ ; YES.
BIS #ERR!HNF!OPI,@RLCS ; NO SET THE ERROR BITS...
BR 7$ ;...AND RETURN.
2$: CMP R5,#BB7 ; CALLED FROM BBS7 TEST ??
BNE 3$ ; SKIP IF NOT.
BIS #ERR!NXM,@RLCS ; YES, SET ERROR BITS...
BR 7$ ;...AND RETURN.
3$: MOV #BUF1,R1 ; STANDARD SRC...
MOV #BUF2,R2 ; ...AND DST...
MOV #DFIFO,R3 ; ...AND DUMMY FIFO POINTERS.
CMP BA16,R1 ; IS BA STANDARD (INTERNAL) ??
BEQ 4$ ; BR IF SO.
MOV #100000,R1 ; POINT TO BUF1 IF EXT MEMORY.
MOV #101000,R2 ; DITTO BUF2 (RECEIVER).
MOV #5$ ERRVEC ; SET TRAP CATCHER.
INC MMRO ;***** KT ON *****
4$: ADD #2,@RLBA ; SIMULATE INC BA...
ADC @RLBAE ; ...OVERFLOWING INTO BAE.
MOV (R1)+,(R3)+ ; MEM => FIFO (256 WORDS).
240
CMP R3,#DFIFO+1000 ;
BLO 4$ ;
MOV #DFIFO,R3 ; RESET FIFO POINTER.
5$: ADD #2,@RLBA ;
ADC @RLBAE ; INC BA.
MOV (R3)+,(R2)+ ; FIFO => MEM (255 WORDS).
240
CMP R3,#DFIFO+776 ;
BLO 7$ ; LOOP
BR 6$ ; ...UNTIL DONE.
6$: BIS #ERR!NXM,@RLCS ; ON NXM TRAP, SET ERROR BITS...
MOV #6$,(SP) ; ...AND CONTINUE.
7$: CLR MMRO ;***** KT OFF *****
MOV #TRAP4 ERRVEC ; RESET TRAP CATCHER.
MOV @RLDA,R0 ;
ADD #6,R0 ; ADJUST FINAL DA...
MOVB R0,@RLDA ; ...LO BYTE ONLY.

```

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 D 10
RLV12 EMULATOR PAGE 47-1

SEQ 0120

3399	037150	013777	003442	144154		MOV	GDCRC3, @RLMP	:	FAKE THE 1ST...
3400	037156	013737	003444	003374		MOV	GDCRC4, E.MP1	:	...AND 2ND CRC WORDS.
3401	037164	052777	000200	144132	78:	BIS	#CRDY, @RLCS	:	SET DONE BIT.
3402	037172	052777	000100	144124		BIT	#INTEN, @RLCS	:	INTERRUPT EXPECTED ??
3403	037200	001410				BEQ	8\$:	NO.
3404	037202					GETPRI	RO		
(3)	037202	04440							
3405	037204	42700	177437			BIC	#^CPRI07, RO	:	MASK PRIORITY BITS. TRAP CSGPRI
3406	037210	020037	003340			CMP	RO, BPRIOR		
3407	037214	002002				BGE	8\$:	BR IF CPU >= RL BR LEVEL
3408	037216	005237	010634			INC	INTFLG	:	OTHERWISE, SET INT RECEIVED FLAG.
3409	037222	000207			8\$:	RTS	PC	:	...AND RETURN.
3410									
3411	037224	000005			DUMMY:	.BLKW	5	:	THESE ARE THE DUMMY REGISTERS.
3412	037236	000400			DFIFO:	.BLKW	256.	:	AND A DUMMY FIFO.

3414
3415
3416
3417 040236 040236
3418 040340
3419
3420
3421
3422 040340
(2)
(2) 040340 040362
(2) 040342 000007
(3) 040344
3423
3424
3425
3426
3427
3428
3429 040344
3430 040344
(4) 040344 000000
(3) 040346 000005
(3) 040350
3431 040350 174400
3432 040352 000160
3433 040354 000200
3434 040356 000000
3435 040360 000003
3436 040362
(3) 040362
3437 040362
3438
3439
3440 000001

... FINALLY, A SMALL PATCH BLOCK...

PATCH: := .+100

...AND THAT'S ALL THERE IS TO IT !!!

LASTAD ; LAST USED ADDRESS.

.EVEN
.WORD TSFREE
.WORD TSSIZE

LSLAST::

... NOW CAN A SINGLE UNIT P-TABLE, SO WE CAN RUN UNDER
... DEFAULT CONDITIONS, AND CALL THE BINARY A .BIC FILE.

BGNSETUP 1
BGNPTAB

: 1 UNIT ONLY...
:...USING THE FOLLOWING DEFAULTS.

.WORD 0
.WORD L10066-. /2-1

L10064:

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

: (0) CSR ADDRESS.
: (2) VECTOR.
: (4) PRIORITY.
: (6) DRIVE (BITS 8,9,10).
: (10) CONTROLLER TYPE (RLV12 W/BAE).

L10066:

ENDSETUP

.END

ADJTN	026000	550	3076#																	
ADLIM	002542	98	116#																	
ADR	= 000020 G	147#																		
AFREG	030455	3175	3224#																	
AFTER	010074	722#	899	902																
ALLREG	027100	3099	3108	3111	3114	3120	3127	3134	3151	3159	3167#									
ASSEMB=	000010	17																		
BA16	010352	554*	771	792#	2207*	2223	2226*	2262*	2342*	2351*	3371									
BA22	010352	555*	774	782	793#	2212*	2222	2227*	2261*	2343*	2352*									
BB7	020416	2215#	3363																	
BCCFBK	003406	249#	958*	964*	967*	970	974													
BCSR	003336	229#	524*	527*	556															
BDDAT	003434	260#	1265*	1266*	1267	1271*	1274*	1279*	1301*	1302*	1303	1329*	1330	1352*						
		1353	1378*	1380	1408*	1409*	1410	1421*	1422*	1423	1450*	1451	1459*	1460						
		1483*	1484	1492*	1493	1520*	1521	1530*	1531	1560*	1561	1566*	1567	1574*						
		1575	1600*	1601*	1602	1607*	1608	1615*	1616	1641*	1642*	1643	1648*	1649						
		1656*	1657	1683*	1684*	1685	1690*	1691	1696*	1697	1704*	1705	1730*	1731*						
		1732	1737*	1741*	1774*	1775	1791*	1792	1836*	1840	1847	1856*	1945*	1946*						
		1951	1953	2031*	2032	2037*	2042	2047*	2048	2053*	2054	2063*	2064	2072						
		2083*	2084	2159*	2160	2168	2319*	2320	2372*	2374	2378*	2381	2386*	2389						
		2394*	2397	2407*	2413	2442*	2445	2448*	2450	2478*	2481	2511*	2514	2521*						
		2522	2524	2526	2652*	2758*	2763*	2765	3104	3107	3110	3122	3133	3157						
		709#	785	1346	2472	2507	2612	2638	2729											
BEFORE	010016	278#	1322	1348	1374	1442	1478	1514												
BEGPAT	003670	3174	3223#																	
BEREG	030434	58	60	100	103	110	147#	181	185	1327	2570	2695								
BIT0	= 000001 G	147#																		
BIT00	== 000001 G	147#																		
BIT01	== 000002 G	147#																		
BIT02	== 000004 G	147#																		
BIT03	== 000010 G	147#																		
BIT04	== 000020 G	147#																		
BIT05	== 000040 G	147#																		
BIT06	== 000100 G	147#																		
BIT07	== 000200 G	147#																		
BIT08	== 000400 G	147#																		
BIT09	== 001000 G	147#																		
BIT1	== 000002 G	56	109	147#	174	176	178	180	184	350	2565	2690								
BIT10	== 002000 G	147#	165	2408																
BIT11	== 004000 G	147#	163	164																
BIT12	== 010000 G	147#	161	162	2512															
BIT13	== 020000 G	147#	159	160	2512	2523	2526													
BIT14	== 040000 G	107	147#	158	2523	2575	2700													
BIT15	== 100000 G	106	147#	157	3140															
BIT2	== 000004 G	108	147#	175	176	179	180	186	351	2560	2685									
BIT3	== 000010 G	147#	177	178	179	180	183	352												
BIT4	== 000020 G	147#	189	353																
BIT5	== 000040 G	147#	354																	
BIT6	== 000100 G	147#	171	187	188	355														
BIT7	== 000200 G	147#	170																	
BIT8	== 000400 G	147#	167	169	356															
BIT9	== 001000 G	147#	168	169	357	2408														
BOE	== 000400 G	147#																		
BPRIOR	003340	230#	529*	568	1838	3406														
BRL	002351	63	75#																	
BUFEND	006364	450#	2030	2983	2986	3003														
BUF1	004364	448#	554	792	808	832	2059	2109	2122	2155	2226	2266	2351	2505						

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 I 10
PAGE 48-3
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0125

DLYCNT	003462	271#	549*	746										
DMA1	016762	1978#												
DMA2	017030	1993#												
DMA3	017076	2008#												
DMODE	024566	2884#	2945*	2948*	2954	2967								
DOIT	025064	2950#												
DRDY =	000001	181#	1266	1302	1409	1422	1601	1642	1684	1731	2379	2387		
DRIVE	003346	233#	530*	768										
DRN	002363	65#	76#											
DROPPE	003464	272#	485*	511	609	611*	621	623*						
DRST =	000010	183#	2435	2444										
DRTIM	U30523	3226#												
DSO	= 000000	166#												
DS1	= 000400	167#	2416											
DS2	= 001000	168#												
DS3	= 001400	169#												
DTABL	024572	2853	2886#											
DUMMY	037224	527	3411#											
EF.CON=	000036	147#	473											
EF.NEW=	000035	147#	471											
EF.PWR=	000034	147#	465											
EF.RES=	000037	147#	469											
EF.STA=	000040	147#	467											
EMURLV	036706	789	3353#											
EMO	030636	586	2238#											
EM1	030706	1135	2239#											
EM10	031303	2034	2249#											
EM100	034574	2769	2249#											
EM100A	034630	2771	2305#											
EM100B	034713	2773	3307#											
EM101	035007	2376	3309#											
EM102	035070	2383	2391	3310#										
EM103	035145	2399	3311#											
EM104	035227	2415	3312#											
EM104X	035235	2402*	2417*	3313#										
EM105	035311	2447	3314#											
EM106	035373	2452	3315#											
EM107	035443	2456	3316#											
EM110	035516	2483	3317#											
EM111	035577	2487	3318#											
EM112	035644	2518	3319#											
EM113	035736	2530	3320#											
EM113A	036010	2528	3321#											
EM114	036066	2534	3322#											
EM115	036137	2620	2644	3323#										
EM115X	036205	2606*	2625*	3324#										
EM116	036333	2634	2625*											
EM117	036313	2652	2626*											
EM117X	036355	2607*	2626*	3327#										
EM118	036403	2737	2626*											
EM119	036454	2558	2683	3329#										
EM119X	036521	2557*	2683	3331#										
EM12	031344	2044	2230#											
EM14	031405	1777	2231#											
EM14A	031440	1785	2232#											
EM14B	031502	1796	2233#											

GGGGG

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 J 10 PAGE 48-4
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0126

EM15	031544	1821	2152	3254#			
EM16	031606	1854	2255#				
EM17	031641	1918	2256#				
EM2	0320733	1157	2240#				
EM20	0321706	2050	2257#				
EM21	0321757	2056	2258#				
EM22	0322037	2070	2166	3259#			
EM23	0322103	2086	2260#				
EM27	0322151	1890	2261#				
EM27X	0322251	1879*	1880*	1895*	1896*	3263#	
EM3	0320760	1179	2241#				
EM31	0322260	1949	2264#				
EM32	0322306	1956	2265#				
EM4	0321005	1201	2242#				
EM4A	0321032	1226	2243#				
EM4B	0321060	1232	2244#				
EM44	0322333	1687	2266#				
EM45	0322366	1693	2267#				
EM46	0322421	1699	2268#				
EM47	0322454	1707	2269#				
EM5	0321127	1305	2245#				
EM50	0322510	1734	2270#				
EM51	0322547	1739	2271#				
EM52	0322573	1743	2272#				
EM6	0321200	1332	2246#				
EM61	0322617	1412	2273#				
EM62	0322665	1425	2274#				
EM63	0322735	1453	2275#				
EM64	0322805	1462	2276#				
EM65	0323053	1486	2277#				
EM66	0323121	1495	2278#				
EM67	0323171	1523	2279#				
EM68	0323240	1533	2280#				
EM69	0323311	1269	2281#				
EM7	0321226	1355	2247#				
EM70	0323345	1275	2282#				
EM71	0323401	1276	2283#				
EM71A	0323435	1281	2284#				
EM72	0323472	1563	2285#				
EM73	0323525	1569	2286#				
EM73A	0323560	1577	2287#				
EM74	0323617	1604	2288#				
EM75	0323651	1610	2289#				
EM75A	0323703	1618	2290#				
EM76	0323737	1645	2291#				
EM77	0323772	1651	2292#				
EM77A	0324025	1659	2293#				
EM8	0321254	1382	2248#				
EM80	0324061	2224	2294#				
EM80X	0324112	2202*	2294*	3295#			
EM90	0324147	2205*	2296#				
EM91	0324223	2310	2297#				
EM92	0324300	2338	2298#				
EM98	0324347	2025	2146	3299#			
EM99	0324404	643	3113	3119	3126	3149	3300#
ENDDAT	004362	444#					

LOT	= 000010	G	147#				
LSI	006574		494	504#			
LSACP	002110	G	42#				
LSAPT	002036	G	42#				
LSAU	007432	G	42#	620#			
LSAUT	002070	G	42#				
LSAUTO	007214	G	42#	577#			
LSCCP	002106	G	42#				
LSCLEA	007272	G	42#	592#			
LSCO	002032	G	42#				
LSDEPO	002011	G	42#				
LSDESC	002122	G	42#	44#			
LSDESP	002076	G	42#				
LSDEVP	002060	G	42#				
LSDISP	026052	G	42#	3087#	3090#		
LSDLY	002116	G	42#				
LSDTP	002040	G	42#				
LSDTYP	002034	G	42#				
LSDU	007350	G	42#	608#			
LSDUT	002072	G	42#				
LSDVTY	002142	G	42#	46#			
LSEF	002052	G	42#				
LSEVI	002044	G	42#				
LSETP	002102	G	42#				
LSEXP1	002046	G	42#				
LSEXP4	002064	G	42#				
LSEXP5	002066	G	42#				
LSHARD	002172	G	42#	55#			
LSHINE	002120	G	42#				
LSHPCP	002016	G	42#				
LSHPTP	002022	G	42#				
LSHW	002376	G	42#	81#			
LSICP	002104	G	42#				
LSINIT	006374	G	42#	464#			
LSLADP	002026	G	42#				
LSLAST	040344	G	37	42	3422#	3437	
LSLOAD	002100	G	42#				
LSLUN	002074	G	42#				
LSMREV	002050	G	42#				
LSNAME	002000	G	42#				
LSPRIO	002042	G	42#				
LSPROT	006366	G	42#	456#			
LSPRT	002112	G	42#				
LSREPP	002062	G	42#				
LSREV	002010	G	42#				
LSRPT	011644	G	42#	1099#			
LSSOFT	002412	G	42#	97#			
LSSPC	002056	G	42#				
LSSPCP	002020	G	42#				
LSSPTP	002024	G	42#				
LSSTA	002030	G	42#				
LSSW	003304	G	42#	135#			
LSTEST	002114	G	42#				
LSTIPL	002014	G	42#				
LSUNIT	002012	G	42#	511	520	578	609
L10000	002266	G	55	66#			

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 C 11
PAGE 48-10
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0132

L10001	002410	81	91#						
L10002	002542	97	112#						
L10003	003322	135	143#						
L10005	007212	573#							
L10006	007270	588#							
L10007	007346	604#							
L10010	007430	616#							
L10011	007506	628#							
L10012	012112	1107	1113#						
L10013	012176	1138#							
L10014	012262	1160#							
L10015	012346	1182#							
L10016	012432	1204#							
L10017	012554	1233#							
L10020	013016	1282#							
L10021	013138	1311#							
L10022	013242	1338#							
L10023	013330	1361#							
L10024	013434	1388#							
L10025	013662	1431#							
L10026	014046	1468#							
L10027	014214	1501#							
L10030	014406	1539#							
L10031	014610	1578#							
L10032	015034	1619#							
L10033	015260	1660#							
L10034	015550	1708#							
L10035	015744	1744#							
L10036	016156	1787	1798#						
L10037	016244	1823#							
L10040	016412	1861#							
L10041	016546	1898#							
L10042	016632	1921#							
L10043	016760	1957#							
L10044	017660	2015	2090#						
L10045	017026	1989#							
L10046	017074	2004#							
L10047	017124	2014#							
L10050	020304	2129	2182#						
L10051	017712	2115#							
L10052	017746	2128#							
L10053	020500	2201	2228#						
L10054	021274	2253	2353#						
L10055	021646	2420#							
L10056	022052	2457#							
L10057	022220	2488#							
L10060	023500	2535#							
L10061	023362	2595	2664#						
L10062	023760	2720	2740#						
L10063	027364	3191#							
L10064	040350	3430#							
L10066	040362	3430	3436#						
MAINT =	000000	173#	1882	1914	1943	2021	2142	2214	2284
MFPT =	000007	499#	500						
MIMIC =	036704	525	726	787	3351#				
PK =	000001	185#	2435	2444	2471	2480			

MMRO = 177572	1006#	1026	1058*	1079*	2249	2271*	2279*	2317*	2324*	3376*	3394*
MMR1 = 177574	1007#										
MMR2 = 177576	1008#										
MMR3 = 172516	1009#	1055	1056*	1063*	2259*						
MPXCLK 003312	139#	2560	2565	2570	2575	2579	2685	2690	2695	2700	2704
MSGNOT 012104	1100*	1103*	1104	1111#							
MSG2 011764	1104	1108#									
MSG3 012014	1105	1109#									
MSG4 012047	1106	1110#									
NEWPAS 006620	472	511#	521								
NLSI 006570	498	503#									
NOANS 025530	2921*	2923	3055#								
NON 030630	647	3236#									
NOOP = 000000	172#										
NOSIG 026160	2558	2683	3097#								
NXM = 020000	159#	662	2027	2148	2216	2300	3365	3391			
NXMMES 030563	664	669	3229#								
NXTU 006662	475	518#	523								
OCTMOD 025324	2884	2945	3012#								
OCTTXT 025770	3012	3067#									
OPI = 002000	165#	665	673	681	686	1794	1885	1947	2027	2148	2833 3360
OPINES 030623	689	3235#									
OPIMN 003446	265#	1953									
OPIMX 003450	266#	1951									
OSAPTS= 000000	17#	42									
OSAU = 000001	17#	41#	42								
OSBGNR= 000001	17#	41#	42								
OSBGNS= 000001	17#	41#	42								
OSDU = 000001	17#	41#	42								
OSERR= 000000	17#	42									
OSGNSW= 000001	17#	41#	42								
OSPOIN= 000001	17#	41#	42								
OSSETU= 000001	17#	41#	42	3422							
PAR = 020000	160#	662									
PARMES 030570	667	3230#									
PAR4 = 172550	2246#	2263*	2329	2344*	2345						
PATCH 040236	3417#										
PATCRC 004200	384#	1979	1994								
PATDAT 004272	416#										
PDSW 003306	137#	480	2893*								
PNT = 001000	147#										
PRGSIZ= 000100	37#										
PRI = 002000	147#										
PRI00 = 000000	147#	1812	1911	2140							
PRI01 = 000040	147#										
PRI02 = 000100	147#										
PRI03 = 000140	147#										
PRI04 = 000200	84	147#	3433								
PRI05 = 000240	147#										
PRI06 = 000300	147#										
PRI07 = 000340	147#	570	572	603	1249	1822	1836	1860	1920	2179	3405
PRM 003047	105	124#									
PRMDMP 003024	103	123#									
PROMID 003310	138#	2764									
PRTPA 026610	3125	3129	3132	3138#							
PSERR 024506	2615	2732	2827*	2833*	2851*	2859#					

G
G
G
G
G
G
G
G
G

SETRAN 010452
SFN = 000006
SIGN = 000004
SIMBCC 011060
SKP1 = 000401
SKP2 = 000402
SKP3 = 000403
SINGLSE 003314
START 006452
STHS = 000100
SVCGBL= 000000

SVCINS= 000001

830#	2009	2264	1157#	1179#	1201#	1226#							
196#	586#	1135#											
186#													
922	930	935	952#										
217#	2748												
218#	502	547	901	2203	2294	2519	3083	3085					
219#													
140#	2550	2591	2675	2716	2846*								
468	478#												
188#													
17#	19#	38#	42	44	46	48#	55	81	97	135	456	464	
577	592	608	620	1099	3090	3096	3422#						
17#	23#	39#	42	44	46	49#	55	56	57	58	59	60	
61	62	63	64	65	66	81	97	98	99	100	101	102	
103	104	105	106	107	108	109	110	111	112	113	114	115	
467	468	469	470	471	472	473	474	509	513	522	465	466	
586	587	588	603	604	612	616	624	628	900	1104	523	573	
1107	1113	1135	1136	1137	1138	1157	1158	1159	1160	1179	1180	1181	
1182	1201	1202	1203	1204	1226	1227	1228	1232	1233	1249	1269	1273	
1276	1281	1282	1294	1305	1306	1310	1311	1323	1332	1333	1337	1338	
1349	1355	1356	1360	1361	1375	1382	1383	1387	1388	1400	1412	1413	
1425	1426	1430	1431	1443	1453	1454	1462	1463	1467	1468	1479	1486	
1487	1495	1496	1500	1501	1515	1523	1524	1533	1534	1538	1539	1563	
1564	1569	1570	1577	1578	1604	1605	1610	1611	1618	1619	1645	1646	
1651	1652	1659	1660	1687	1688	1693	1694	1699	1700	1707	1708	1734	
1735	1739	1740	1743	1744	1777	1778	1785	1786	1787	1796	1797	1798	
1812	1821	1822	1823	1839	1840	1854	1855	1859	1860	1861	1890	1891	
1898	1911	1918	1919	1920	1921	1949	1956	1957	1978	1989	1993	2004	
2008	2014	2015	2019	2025	2026	2029	2034	2035	2044	2045	2050	2051	
2056	2057	2070	2072	2073	2081	2086	2087	2090	2106	2115	2119	2128	
2129	2133	2140	2146	2147	2152	2153	2166	2168	2169	2177	2178	2179	
2182	2201	2224	2228	2233	2263	2303	2310	2338	2340	2341	2353	2376	
2282	2291	2309	2315	2320	2347	2352	2356	2357	2348	2349	2353	2318	
2328	2330	2334	2335	2338	2350	2395	2401	2456	2483	2487	2488	2518	
2528	2530	2534	2535	2538	2558	2595	2601	2620	2644	2644	2655	2664	
2683	2720	2724	2737	2738	2740	2769	2771	2771	2845	3012	3049	3051	
3090	3101	3104	3107	3110	3113	3116	3119	3122	3126	3133	3142	3149	
3150	3157	3158	3161	3173	3174	3175	3176	3181	3182	3188	3189	3190	
3191	3404	1978	1993	2008	2106	2119	588	604	616	628	1113	1120	
17#	21#	66	91	112	143	573	1169	1186	1191	1204	1208	1211	
1125	1138	1142	1147	1160	1164	1310	1311	1315	1320	1337	1338	1342	
1233	1237	1247	1282	1286	1291	1388	1392	1397	1430	1431	1435	1440	
1376	1360	1361	1365	1370	1387	1388	1510	1538	1539	1543	1551	1578	
1467	1468	1472	1476	1500	1501	1505	1670	1708	1712	1717	1744	1748	
1582	1588	1619	1623	1629	1660	1664	1670	1708	1865	1870	1898	1903	
1754	1798	1802	1809	1833	1827	1834	1859	1861	1865	1870	1898	1903	
1907	1921	1925	1935	1957	1961	1973	1989	2004	2014	2087	2090	2094	
2101	2115	2128	2178	2182	2186	2195	2228	2232	2245	2341	2353	2358	
2366	2420	2424	2427	2457	2461	2463	2488	2492	2494	2535	2539	2545	
2723	2664	2668	2670	2738	2740	3191	3430	3436					
2662	20#	1126	1148	1170	1192	1212	1248	1292	1321	1347	1371	1398	
17#	1477	1511	1552	1589	1630	1671	1718	1755	1810	1835	1871	1908	
1936	1974	2102	2196	2247	2267	2428	2464	2495	2546	2671	1138#	1160#	
17#	66#	91#	112#	143#	173#	238#	264#	266#	268#	267#	1113#	1160#	
1182#	1204#	1233#	1282#	1294#	1311#	1323#	1338#	1349#	1361#	1375#	1388#	1400#	
1431#	1443#	1468#	1479#	1501#	1515#	1539#	1578#	1619#	1660#	1708#	1744#	1798#	

SVCSUB= 000001
SVCTAG= 000001

SVCTST= 000001

SLSYM= 010000

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 306(1063) 08-FEB-82 10:07 PAGE 48-14
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0136

	1823#	1839#	1861#	189#	1921#	1957#	1989#	2004#	2014#	2019#	2090#	2115#	2128#
TAB 025756	2133#	2182#	2228#	2265#	2353#	2420#	2457#	2488#	2535#	2601#	2664#	2724#	2740#
TCLK0 036536	2966#	3064#											
TCLK1 036554	2563#	2688#	3334#										
TCLK2 036567	2568#	2693#	3335#										
TCSR 003344	2573#	2698#	3336#										
	2758#	2782#	3080	538*	2370*	2371*	2372	2384*	2392*	2400*	2406	2750	2753*
TCSUO 002776	102	122#											
TCSO 003320	142#	535	2909										
TEMHI 003460	270#	516*	553										
TEMLO 003456	269#	515*	552	2010									
TEMP1 003422	255#	917*	918	920*	921	928*	929						
TEMP2 003412	251#	955*	978*										
TEMP3 003414	252#	956*	960*										
TEMP4 003416	253#	957*	959	969*	971	976*	977*	980					
TEMP5 003420	254#	918*	919*	920	927*	928							
TLMBP 002575	99	118#											
TLMF 003316	141#	533											
TLMIN 002751	100	121#											
TLMOK 023762	2368	2429	2465	2496	2747#								
TLMPID 023766	2547	2672	2749#										
TLM1 021276	2367#												
TLM2 021650	2428#												
TLM3 022054	2464#												
TLM4 022222	2495#												
TLM5 022502	2546#												
TLM6 023364	2671#												
TMPO 003424	256#	807*	810	813*	1820*	1837*	1857*	2038*	2039*	2040*	2041	2066*	2072
	2162*	2168	2327*	2653*	2950*	2953	3116	3133	3157	2040*	2041	2066*	2072
TMP1 003426	257#	2067*	2072	2163*	2168	2222*	2296*	2330*	2332*	2337*	2455*	2486*	2533*
	2619*	2643*	2654*	2736*	3141*	3142	3149	3157					
TMP2 003430	258#	2058*	2068	2071*	2079	2081	2154*	2164	2167*	2175	2177	2223*	2297*
	2329*	2331*	2336*	2563*	2568*	2573*	2688*	2693*	2698*	3138	3140*	3142	3161
TPEAK 036602	2577*	2582*	2702*	2707*	3161	3337#							
TRAP4 010624	569	879#	2280	3395									
TRPFLG 010626	580*	583	880#	1127*	1132	1149*	1154	1171*	1176	1193*	1198	1215*	1224
	1230	2248*	2251	2270*	2276*	2298	2752*	2755					
TSEC 036523	3161	3332#											
TSECK 036532	2557	2682	2821*	2825*	2831*	2837*	2841*	2849*	3333#				
TURKY 024510	2845	2861#											
TSARGC= 000001	42#	612#	624#	1104#	1105#	1106#	2072#	2081#	2168#	2177#	2845#	3012#	3051#
	3101#	3104#	3107#	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#
	3157#	3158#	3161#	3173#	3174#	3175#	3176#	3188#					
TSCODE= 004032	56#	57#	58#	59#	60#	61#	62#	63#	64#	65#	98#	99#	100#
	101#	102#	103#	104#	105#	106#	107#	108#	109#	110#	111#		
TSERRN= 000264	17#	526#	900#	1135#	1157#	1179#	1201#	1226#	1232#	1269#	1273#	1276#	1281#
	1305#	1332#	1355#	1382#	1412#	1425#	1453#	1462#	1486#	1495#	1523#	1533#	1563#
	1569#	1577#	1604#	1610#	1618#	1645#	1651#	1659#	1687#	1693#	1699#	1707#	1734#
	1739#	1743#	1777#	1785#	1796#	1821#	1854#	1890#	1918#	1949#	1956#	2025#	2034#
	2044#	2050#	2056#	2070#	2086#	2146#	2152#	2166#	2224#	2305#	2310#	2338#	2376#
	2583#	2591#	2599#	2415#	2447#	2452#	2456#	2483#	2487#	2518#	2528#	2530#	2534#
	2558#	2620#	2634#	2644#	2655#	2683#	2737#	2769#	2771#	2773#			
TSEXCP= 000000	61#	62#	63#	65#	68#	102#	105#	111#					
TSFLAG= 000040	1107#	1306#	1333#	1356#	1383#	1413#	1426#	1454#	1463#	1487#	1496#	1524#	1534#

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 M 11
CROSS REFERENCE TABLE -- USER SYMBOLS PAGE 48-15

SEQ 0137

TSFREE= 040362	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#						
TSGMAN= 000000	3422#	3437#												
TSHILI= 000006	17#													
TSLAST= 000001	61#	62#	63#	65#	98#	102#	105#	111#						
TSLOLI= 000000	17#	3422#	3429#											
TLSYM= 010000	61#	62#	63#	65#	98#	102#	105#	111#						
	17#	66	91	112	143	573	588	604	616	628	1113	1138	1160	
	1182	1204	1233	1282	1311	1338	1361	1388	1431	1468	1501	1539	1578	
	1619	1660	1708	1744	1798	1823	1861	1898	1921	1957	1989	2004	2014	
	2090	2115	2128	2182	2228	2353	2420	2457	2488	2535	2664	2740	3191	
TSLTNO= 000043	3422#													
TSNEST= 177777	17#	55#	57	59	64	66#	81#	91#	97#	101	104	112#	135#	
	143#	456#	460#	464#	573#	577#	588#	592#	604#	608#	616#	620#	628#	
	1099#	1113#	1126#	1138#	1148#	1160#	1170#	1182#	1192#	1204#	1212#	1233#	1248#	
	1282#	1292#	1294#	1310#	1311#	1321#	1323#	1337#	1338#	1347#	1349#	1360#	1361#	
	1371#	1375#	1387#	1388#	1398#	1400#	1430#	1431#	1441#	1443#	1467#	1468#	1477#	
	1479#	1500#	1501#	1511#	1515#	1538#	1539#	1552#	1578#	1589#	1619#	1630#	1660#	
	1671#	1708#	1718#	1744#	1755#	1798#	1810#	1823#	1835#	1839#	1859#	1861#	1871#	
	1898#	1908#	1921#	1936#	1957#	1974#	1978#	1989#	1993#	2004#	2008#	2014#	2019#	
	2087#	2090#	2102#	2106#	2115#	2119#	2128#	2133#	2178#	2182#	2008#	2014#	2019#	
	2265#	2341#	2353#	2367#	2420#	2428#	2457#	2464#	2488#	2495#	2535#	2546#	2601#	
	2662#	2664#	2671#	2724#	2738#	2740#	3096#	3191#						
TSNSO = 000011	55#	57	59	64	66	81#	91	97#	101	104	112	135#	143	
	456#	460	464#	573	577#	588	592#	604	608#	616	620#	628	1099#	
	1113	1126#	1138	1148#	1160	1170#	1182	1192#	1204	1212#	1233	1248#	1282	
	1292#	1311	1321#	1338	1347#	1361	1371#	1388	1398#	1431	1441#	1468	1477#	
	1501	1511#	1539	1552#	1578	1589#	1619	1630#	1660	1671#	1708	1718#	1744	
	1755#	1798	1810#	1823	1835#	1861	1871#	1898	1908#	1921	1936#	1957	1974#	
	2090	2102#	2182	2196#	2228	2247#	2353	2367#	2420	2428#	2457	2464#	2488	
TSNS1 = 000003	2495#	2535	2546#	2664	2671#	2740	3096#	3191						
	1294#	1310	1323#	1337	1349#	1360	1375#	1387	1400#	1430	1443#	1467	1479#	
	1500	1515#	1538	1839#	1859	1978#	1989	1993#	2004	2008#	2014	2019#	2087	
	2106#	2115	2119#	2128	2133#	2178	2265#	2341	2601#	2662	2724#	2738		
TSPCNT= 000000	3429#	3430#												
TSPTAB= 010065	3430#													
TSPTHV= 000001	42	3437#												
TSPTNU= 000001	17#	3430#	3437											
TSSAVL= 177777	17#													
TSSEGL= 177777	17#	1294#	1306	1310#	1323#	1333	1337#	1349#	1356	1360#	1375#	1383	1387#	
	1400#	1413	1426	1430#	1443#	1454	1463	1467#	1479#	1487	1496	1500#	1515#	
	1524	1534	1538#	1839#	1859	2019#	2029	2087#	2133#	2178#	2265#	2341#	2601#	
TSSEKO= 010000	2662#	2724#	2738#											
	1294#	1306	1310	1323#	1333	1337	1349#	1356	1360	1375#	1383	1387	1400#	
	1413	1426	1430	1443#	1454	1463	1467	1479#	1487	1496	1500	1515#	1524	
	1534	1538	1839#	1859	2019#	2029	2087	2133#	2178	2265#	2341	2601#	2662	
	2724#	2738												
TSSIZE= 000007	3422	3437#												
TSSUBN= 000000	17#	1126#	1148#	1170#	1192#	1212#	1248#	1292#	1321#	1347#	1371#	1398#	1441#	
	1477#	1511#	1552#	1589#	1630#	1671#	1718#	1755#	1810#	1835#	1871#	1908#	1936#	
	1974#	1978#	1993#	2008#	2102#	2106#	2119#	2196#	2247#	2367#	2428#	2464#	2495#	
	2546#	2671#												
TSTAGL= 177777	17#													
TSTAGN= 010067	17#	55#	81#	97#	135#	456#	464#	577#	592#	608#	620#	1099#	1126#	
	1148#	1170#	1192#	1212#	1248#	1292#	1321#	1347#	1371#	1398#	1441#	1477#	1511#	
	1552#	1589#	1630#	1671#	1718#	1755#	1810#	1835#	1871#	1908#	1936#	1974#	1978#	
	1993#	2008#	2102#	2106#	2119#	2196#	2247#	2367#	2428#	2464#	2495#	2546#	2671#	

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 PAGE 48-17
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0139

TSSRPT= 010012	1099#	1107	1113												
TSSSEG= 010000	1294#	1306	1310#	1323#	1333	1337#	1349#	1356	1360#	1375#	1383	1387#	1400#		
	1413	1426	1430#	1443#	1454	1463	1467#	1479#	1487	1496	1500#	1515#	1524		
	1534	1538#	1839#	1859#	2019#	2029	2087#	2133#	2178#	2265#	2341#	2601#	2662#		
	2724#	2738#													
TSSSOE= 010002	97#	112													
TSSSUB= 010052	1978#	1989	1993#	2004	2008#	2014	2106#	2115	2119#	2128					
TSSSW = 010003	135#	143													
TSSTES= 010062	1126#	1138	1148#	1160	1170#	1182	1192#	1204	1212#	1233	1248#	1282	1292#		
	1311	1321#	1338	1347#	1361	1371#	1388	1398#	1431	1441#	1468	1477#	1501		
	1511#	1539	1552#	1578	1589#	1619	1630#	1660	1671#	1708	1718#	1744	1755#		
	1787	1798	1810#	1823	1835#	1861	1871#	1898	1908#	1921	1936#	1957	1974#		
	2015	2090	2102#	2129	2182	2196#	2201	2228	2247#	2253	2353	2367#	2420		
	2428#	2457	2464#	2488	2495#	2535	2546#	2595	2664	2671#	2720	2740			
T1	012114	G													
T10	013332	G													
T11	013436	G													
T12	013664	G													
T13	014050	G													
T14	014216	G													
T15	014410	G													
T16	014612	G													
T17	015036	G													
T18	015262	G													
T19	015552	G													
T2	012200	G													
T20	015746	G													
T21	016160	G													
T22	016246	G													
T23	016414	G													
T24	016550	G													
T25	016634	G													
T26	016762	G													
T26.1	016762														
T26.2	017030														
T26.3	017076														
T27	017662	G													
T27.1	017662														
T27.2	017714														
T28	020306	G													
T29	020502	G													
T3	012264	G													
T30	021276	G													
T31	021650	G													
T32	022054	G													
T33	022222	G													
T34	022502	G													
T35	023364	G													
T4	012350	G													
T5	012434	G													
T6	012536	G													
T7	013020	G													
T8	013140	G													
T9	013244	G													
UAM = 000200	G														
UNITST = 003322	G														

517* 518* 520 522 536 578 587 1136 1158 1180 1202 1227

BCOMPL	466	468	470	472	474	3182									
BEGIN.	211#	1126	1148	1170	1192	1212	1248	1292	1321	1347	1371	1398	1441	1477	1511
	1552	1589	1630	1671	1718	1755	1810	1835	1871	1908	1936	1974	2102	2196	2247
	2367	2428	2464	2495	2546	2671									
BGNAU	620														
BGNAUT	577														
BGNCLN	592														
BGNDU	608														
BGNHRD	55														
BGNHW	81														
BGNINI	464														
BGNMSG	3096														
BGNPRO	456														
BGNPTA	3430														
BGNRPT	1099														
BGNSEG	1294	1323	1349	1375	1400	1443	1479	1515	1839	2019	2133	2265	2601	2724	
BGNSET	3429														
BGNSFT	97														
BGNSUB	1978	1993	2008	2106	2119										
BGNSW	135														
BGNTST	1126	1148	1170	1192	1212	1248	1292	1321	1347	1371	1398	1441	1477	1511	1552
	1589	1630	1671	1718	1755	1810	1835	1871	1908	1936	1974	2102	2196	2247	2367
	2428	2464	2495	2546	2671										
BNCOMP	523														
CKLOOP	1564	1570	1605	1611	1646	1652	1688	1694	1700	1735	1740	1778	1786	1797	1855
	1891	1919	2026	2035	2045	2051	2057	2073	2147	2153	2169	2340			
DESCRI	44														
DEVTYP	46														
DFERR	204#	900	1232	1269	1273	1276	1281	1305	1332	1355	1382	1412	1425	1453	1462
	1486	1495	1523	1533	1563	1569	1577	1604	1610	1618	1645	1651	1659	1687	1693
	1699	1707	1734	1739	1743	1777	1785	1796	1821	1854	1890	1918	1949	1956	2025
	2034	2044	2050	2056	2070	2086	2146	2152	2166	2224	2305	2310	2338	2376	2383
	2391	2399	2415	2447	2452	2456	2483	2487	2518	2528	2530	2534	2558	2620	2634
	2644	2655	2683	2737	2769	2771	2773								
DISPAT	3090														
DISPLA	99														
DOCLN	513	1137	1159	1181	1203	1228	3049	3190							
DODU	587	1136	1158	1180	1202	1227	3189								
DORPT	509														
ENDAU	628														
ENDAUT	588														
ENDCLN	604														
ENDDU	616														
ENDHRD	66														
ENDHW	91														
ENDINI	573														
ENDMSG	3191														
ENDPRO	460														
ENDPTA	3436														
ENDRPT	1113														
ENDSEG	1310	1337	1360	1387	1430	1467	1500	1538	1859	2087	2178	2341	2662	2738	
ENDSET	3437														
ENDSFT	112														
ENDSUB	1989	2004	2014	2115	2128										
ENDSW	143														
ENDTST	1138	1160	1182	1204	1233	1282	1311	1338	1361	1388	1431	1468	1501	1539	1578

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 M 11
CROSS REFERENCE TABLE -- MACRO NAMES PAGE 49-1

SEQ 0142

	1619	1660	1708	1744	1798	1823	1861	1898	1921	1957	2090	2182	2228	2353	2420
EQUALS	2457	2488	2535	2664	2740										
ERRDF	147														
	900	1232	1269	1273	1276	1281	1305	1332	1355	1382	1412	1425	1453	1462	1486
	1495	1523	1533	1563	1569	1577	1604	1610	1618	1645	1651	1659	1687	1693	1699
	1707	1734	1739	1743	1777	1785	1796	1821	1854	1890	1918	1949	1956	2025	2034
	2044	2050	2056	2070	2086	2146	2152	2166	2224	2305	2310	2338	2376	2383	2391
	2399	2415	2447	2452	2456	2483	2487	2518	2528	2530	2534	2558	2620	2634	2644
ERRSF	2655	2683	2737	2769	2771	2773									
ESCAPE	586	1135	1157	1179	1201	1226									
EXIT	1306	1333	1356	1383	1413	1426	1454	1463	1487	1496	1524	1534			
GETPRI	1107	1787	2015	2029	2129	2201	2253	2595	2720						
GPHARD	3404														
GPRMA	522														
GPRMD	61	62	102												
GPRML	63	65	98	105	111										
HEADER	56	58	60	100	103	106	107	108	109	110					
INLOOP	42														
LASTAD	3181														
MSBYTE	3422														
MSCHEC	42#														
MSCNTO	1107#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#						
	56#	58#	60#	61#	62#	63#	65#	98#	100#	102#	103#	105#	106#	107#	108#
MSCOUN	109#	110#	111#												
	612#	624#	1104#	1105#	1106#	2072#	2081#	2168#	2177#	2845#	3012#	3051#	3101#	3104#	3107#
	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#	3157#	3158#	3161#	3173#	3174#
	3175#	3176#	3188#												
MSDATA	42#	44#	46#												
MSDECR	66#	91#	112#	143#	460#	573#	588#	604#	616#	628#	1113#	1138#	1160#	1182#	1204#
	1233#	1282#	1310#	1311#	1337#	1338#	1360#	1361#	1387#	1388#	1430#	1431#	1467#	1468#	1500#
	1501#	1538#	1539#	1578#	1619#	1660#	1708#	1744#	1798#	1823#	1859#	1861#	1898#	1921#	1957#
	1989#	2004#	2014#	2087#	2090#	2115#	2128#	2178#	2182#	2228#	2341#	2353#	2420#	2457#	2488#
MSDEFA	2535#	2662#	2664#	2738#	2740#	3191#	3430#								
	56#	58#	60#	61#	62#	63#	65#	98#	100#	102#	103#	105#	106#	107#	108#
MSSENDE	109#	110#	111#												
	66#	91#	112#	143#	573#	588#	604#	616#	628#	1113#	1138#	1160#	1182#	1204#	1233#
	1282#	1310#	1311#	1337#	1338#	1360#	1361#	1387#	1388#	1430#	1431#	1467#	1468#	1500#	1501#
	1538#	1539#	1578#	1619#	1660#	1708#	1744#	1798#	1823#	1859#	1861#	1898#	1921#	1957#	1989#
	2004#	2014#	2087#	2090#	2115#	2128#	2178#	2182#	2228#	2341#	2353#	2420#	2457#	2488#	2535#
MSERRI	2662#	2664#	2738#	2740#	3191#										
	586#	900#	1135#	1157#	1179#	1201#	1226#	1232#	1269#	1273#	1276#	1281#	1305#	1332#	1355#
	1382#	1412#	1425#	1453#	1462#	1486#	1495#	1523#	1533#	1563#	1569#	1577#	1604#	1610#	1618#
	1645#	1651#	1659#	1687#	1693#	1699#	1707#	1734#	1739#	1743#	1777#	1785#	1796#	1821#	1854#
	1890#	1918#	1949#	1956#	2025#	2034#	2044#	2050#	2056#	2070#	2086#	2146#	2152#	2166#	2224#
	2305#	2310#	2338#	2376#	2383#	2391#	2399#	2415#	2447#	2452#	2456#	2483#	2487#	2518#	2528#
	2530#	2534#	2558#	2620#	2634#	2644#	2655#	2683#	2737#	2769#	2771#	2773#			
MSERCA	1306#	1333#	1356#	1383#	1413#	1426#	1454#	1463#	1487#	1496#	1524#	1534#			
MSERCS	1306#	1333#	1356#	1383#	1413#	1426#	1454#	1463#	1487#	1496#	1524#	1534#			
MSXCP	61#	62#	63#	65#	98#	102#	105#	111#							
MSXIT	1107#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#						
MSXSE	1107#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#						
MSXTJ	1107#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#						
MSGEN	42#	44#	46#	55#	66#	81#	91#	97#	112#	135#	143#	456#	464#	573#	577#
	588#	592#	604#	608#	616#	620#	628#	1099#	1113#	1126#	1138#	1148#	1160#	1170#	1182#
	1192#	1204#	1212#	1233#	1248#	1282#	1292#	1310#	1311#	1321#	1337#	1338#	1347#	1360#	1361#
	1371#	1387#	1388#	1398#	1430#	1431#	1441#	1467#	1468#	1477#	1500#	1501#	1511#	1538#	1539#

	1552#	1578#	1589#	1619#	1630#	1660#	1671#	1708#	1718#	1744#	1755#	1798#	1810#	1823#	1835#
	1859#	1861#	1871#	1898#	1908#	1921#	1936#	1957#	1974#	1978#	1989#	1993#	2004#	2008#	2014#
	2087#	2090#	2102#	2106#	2115#	2119#	2128#	2178#	2182#	2196#	2228#	2247#	2341#	2353#	2367#
	2420#	2428#	2457#	2464#	2488#	2495#	2535#	2546#	2662#	2664#	2671#	2738#	2740#	3090#	3096#
MSGETS	3191#	3432#	3430#	3436#											
	57#	59#	64#	66#	91#	101#	104#	112#	143#	460#	573#	588#	604#	616#	628#
	1113#	1138#	1160#	1182#	1204#	1233#	1282#	1306#	1310#	1311#	1333#	1337#	1338#	1356#	1360#
	1361#	1383#	1387#	1388#	1413#	1426#	1430#	1431#	1454#	1463#	1467#	1468#	1487#	1496#	1500#
	1501#	1524#	1534#	1538#	1539#	1578#	1619#	1660#	1708#	1744#	1798#	1823#	1859#	1861#	1898#
	1921#	1957#	1989#	2004#	2014#	2029#	2087#	2090#	2115#	2128#	2178#	2182#	2228#	2341#	2353#
MSGETT	2420#	2457#	2488#	2535#	2662#	2664#	2738#	2740#	3191#						
	57#	59#	64#	101#	104#	1107#	1306#	1333#	1356#	1383#	1413#	1426#	1454#	1463#	1487#
MSGNGB	1496#	1524#	1534#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#				
	42#	44#	46#	55#	81#	97#	135#	456#	464#	577#	592#	608#	620#	1099#	3090#
MSGNIN	3096#	3422#													
	42#	44#	46#	55#	56#	57#	58#	59#	60#	61#	62#	63#	64#	65#	66#
	81#	97#	98#	99#	100#	101#	102#	103#	104#	105#	106#	107#	108#	109#	110#
	111#	112#	135#	465#	466#	467#	468#	469#	470#	471#	472#	473#	474#	509#	513#
	522#	523#	573#	586#	587#	588#	603#	604#	612#	616#	624#	628#	900#	1104#	1105#
	1106#	1107#	1113#	1135#	1136#	1137#	1138#	1157#	1158#	1159#	1160#	1179#	1180#	1181#	1182#
	1201#	1202#	1203#	1204#	1226#	1227#	1228#	1232#	1233#	1249#	1269#	1273#	1276#	1281#	1282#
	1294#	1303#	1306#	1310#	1311#	1323#	1332#	1333#	1337#	1338#	1349#	1355#	1356#	1360#	1361#
	1375#	1382#	1383#	1387#	1388#	1400#	1412#	1413#	1425#	1426#	1430#	1431#	1443#	1453#	1454#
	1462#	1463#	1467#	1468#	1479#	1486#	1487#	1495#	1496#	1500#	1501#	1515#	1523#	1524#	1533#
	1534#	1538#	1539#	1563#	1564#	1569#	1570#	1577#	1578#	1604#	1605#	1610#	1611#	1618#	1619#
	1645#	1646#	1651#	1652#	1659#	1660#	1687#	1688#	1693#	1694#	1699#	1700#	1707#	1708#	1734#
	1735#	1739#	1740#	1743#	1744#	1777#	1778#	1785#	1786#	1787#	1796#	1797#	1798#	1812#	1821#
	1822#	1823#	1839#	1840#	1854#	1855#	1859#	1860#	1861#	1890#	1891#	1898#	1911#	1918#	1919#
	1920#	1921#	1949#	1956#	1957#	1978#	1989#	1993#	2004#	2008#	2014#	2015#	2019#	2025#	2026#
	2029#	2034#	2035#	2044#	2045#	2050#	2051#	2056#	2057#	2070#	2072#	2073#	2081#	2086#	2087#
	2090#	2106#	2115#	2119#	2128#	2129#	2133#	2140#	2146#	2147#	2152#	2153#	2166#	2168#	2169#
	2177#	2178#	2179#	2182#	2201#	2224#	2228#	2253#	2265#	2305#	2310#	2338#	2340#	2341#	2353#
	2376#	2383#	2391#	2399#	2415#	2420#	2447#	2452#	2456#	2457#	2483#	2487#	2488#	2518#	2528#
	2530#	2534#	2535#	2558#	2595#	2601#	2620#	2634#	2644#	2655#	2662#	2664#	2683#	2720#	2724#
	2737#	2738#	2740#	2769#	2771#	2773#	2845#	3012#	3049#	3051#	3090#	3101#	3104#	3107#	3110#
	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#	3157#	3158#	3161#	3173#	3174#	3175#
MSGNLS	3176#	3181#	3182#	3188#	3189#	3190#	3191#	3404#	3422#	3430#					
MSGNSU	1310#	1337#	1360#	1387#	1430#	1467#	1500#	1538#	1859#	2087#	2178#	2341#	2662#	2738#	
MSGNTA	1978#	1993#	2008#	2106#	2119#										
	66#	91#	112#	143#	573#	588#	604#	616#	628#	1113#	1138#	1160#	1182#	1204#	1233#
	1282#	1311#	1338#	1361#	1388#	1431#	1468#	1501#	1539#	1578#	1619#	1660#	1708#	1744#	1798#
	1823#	1861#	1898#	1921#	1957#	1989#	2004#	2014#	2090#	2115#	2128#	2182#	2228#	2353#	2420#
MSGNTE	2457#	2488#	2535#	2664#	2740#	3191#	3430#	3436#							
	1126#	1148#	1170#	1192#	1212#	1248#	1292#	1321#	1347#	1371#	1398#	1441#	1477#	1511#	1552#
	1589#	1630#	1671#	1718#	1755#	1810#	1835#	1871#	1908#	1936#	1974#	2102#	2196#	2247#	2367#
MSHAPT	2428#	2464#	2495#	2546#	2671#										
MSHMAP	42#														
MSINCR	42#														
	55#	81#	97#	135#	456#	464#	465#	467#	469#	471#	473#	509#	513#	522#	573#
	577#	586#	587#	588#	592#	603#	604#	608#	612#	616#	620#	624#	628#	900#	1099#
	1104#	1105#	1106#	1113#	1126#	1135#	1136#	1137#	1138#	1148#	1157#	1158#	1159#	1160#	1170#
	1179#	1180#	1181#	1182#	1192#	1201#	1202#	1203#	1204#	1212#	1226#	1227#	1228#	1232#	1233#
	1248#	1249#	1269#	1273#	1276#	1281#	1282#	1292#	1294#	1305#	1306#	1310#	1311#	1321#	1333#
	1332#	1333#	1337#	1358#	1347#	1349#	1355#	1356#	1360#	1361#	1371#	1375#	1382#	1383#	1387#
	1388#	1398#	1400#	1412#	1413#	1425#	1426#	1430#	1431#	1441#	1443#	1453#	1454#	1462#	1463#
	1467#	1468#	1477#	1479#	1486#	1487#	1495#	1496#	1500#	1501#	1511#	1515#	1523#	1524#	1533#

	1534#	1538#	1539#	1552#	1563#	1564#	1569#	1570#	1577#	1578#	1589#	1604#	1605#	1610#	1611#
	1618#	1619#	1630#	1645#	1646#	1651#	1652#	1659#	1660#	1671#	1687#	1688#	1693#	1694#	1699#
	1700#	1707#	1708#	1718#	1734#	1735#	1739#	1740#	1743#	1744#	1755#	1777#	1778#	1785#	1786#
	1787#	1796#	1797#	1798#	1810#	1812#	1821#	1822#	1823#	1835#	1839#	1840#	1854#	1855#	1859#
	1860#	1861#	1871#	1890#	1891#	1898#	1908#	1911#	1918#	1919#	1920#	1921#	1936#	1949#	1956#
	1957#	1974#	1978#	1989#	1993#	2004#	2008#	2014#	2015#	2019#	2025#	2026#	2029#	2034#	2035#
	2044#	2045#	2050#	2051#	2056#	2057#	2070#	2072#	2073#	2081#	2086#	2087#	2090#	2102#	2106#
	2115#	2119#	2128#	2129#	2133#	2140#	2146#	2147#	2152#	2153#	2166#	2168#	2169#	2177#	2178#
	2179#	2182#	2196#	2201#	2224#	2228#	2247#	2253#	2265#	2305#	2310#	2338#	2340#	2341#	2353#
	2367#	2376#	2383#	2391#	2399#	2415#	2420#	2428#	2447#	2452#	2456#	2457#	2464#	2483#	2487#
	2488#	2495#	2518#	2528#	2530#	2534#	2535#	2546#	2558#	2595#	2601#	2620#	2634#	2644#	2655#
	2662#	2664#	2671#	2683#	2720#	2724#	2737#	2738#	2740#	2769#	2771#	2773#	2845#	3012#	3049#
	3051#	3096#	3101#	3104#	3107#	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#
MSLRO	3157#	3158#	3161#	3173#	3174#	3175#	3176#	3181#	3188#	3189#	3190#	3191#	3404#	3429#	3430#
	465#	467#	469#	471#	473#	522#	587#	603#	1136#	1158#	1180#	1202#	1227#	1249#	1812#
	1822#	1840#	1860#	1911#	1920#	2140#	2179#	3169#							
MSMCHI	17#														
MSMCLO	17#														
MSPOP	66#	91#	112#	143#	460#	573#	588#	604#	616#	628#	1113#	1138#	1160#	1182#	1204#
	1233#	1282#	1310#	1311#	1337#	1338#	1360#	1361#	1387#	1388#	1430#	1431#	1467#	1468#	1500#
	1501#	1538#	1539#	1578#	1619#	1660#	1708#	1744#	1798#	1823#	1859#	1861#	1898#	1921#	1957#
	1989#	2004#	2014#	2087#	2090#	2115#	2128#	2178#	2182#	2228#	2341#	2353#	2420#	2457#	2488#
MSPRIN	2535#	2662#	2664#	2738#	2740#	3191#									
	612#	624#	1104#	1105#	1106#	2072#	2081#	2168#	2177#	2845#	3012#	3051#	3101#	3104#	3107#
	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#	3157#	3158#	3161#	3173#	3174#
	3175#	3176#	3188#												
MSPUSH	55#	81#	97#	135#	456#	464#	577#	592#	608#	620#	1099#	1126#	1148#	1170#	1192#
	1212#	1248#	1292#	1294#	1321#	1323#	1347#	1349#	1371#	1375#	1398#	1400#	1441#	1443#	1477#
	1479#	1511#	1515#	1552#	1589#	1630#	1671#	1718#	1755#	1810#	1835#	1839#	1871#	1908#	1936#
	1974#	1978#	1993#	2008#	2019#	2102#	2106#	2119#	2133#	2196#	2247#	2265#	2367#	2428#	2464#
MSPUT	2495#	2546#	2601#	2671#	2724#	3096#									
	612#	624#	1104#	1105#	1106#	2072#	2081#	2168#	2177#	2845#	3012#	3051#	3101#	3104#	3107#
	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#	3157#	3158#	3161#	3173#	3174#
	3175#	3176#	3188#												
MSPUT1	612#	624#	1104#	1105#	1106#	2072#	2081#	2168#	2177#	2845#	3012#	3051#	3101#	3104#	3107#
	3110#	3113#	3116#	3119#	3122#	3126#	3133#	3142#	3149#	3150#	3157#	3158#	3161#	3173#	3174#
	3175#	3176#	3188#												
MSRADI	56#	58#	60#	61#	62#	63#	65#	98#	100#	102#	103#	105#	106#	107#	108#
	109#	110#	111#												
MSRNRO	522#	3404#													
MSSETS	55#	81#	97#	135#	456#	464#	577#	592#	608#	620#	1099#	1126#	1148#	1170#	1192#
	1212#	1248#	1292#	1294#	1321#	1323#	1347#	1349#	1371#	1375#	1398#	1400#	1441#	1443#	1477#
	1479#	1511#	1515#	1552#	1589#	1630#	1671#	1718#	1755#	1810#	1835#	1839#	1871#	1908#	1936#
	1974#	1978#	1993#	2008#	2019#	2102#	2106#	2119#	2133#	2196#	2247#	2265#	2367#	2428#	2464#
MSSVC	2495#	2546#	2601#	2671#	2724#	3096#									
	465#	467#	469#	471#	473#	509#	513#	522#	573#	586#	587#	588#	603#	604#	612#
	616#	624#	628#	900#	1104#	1105#	1106#	1107#	1113#	1135#	1136#	1137#	1138#	1157#	1158#
	1159#	1160#	1179#	1180#	1181#	1182#	1201#	1202#	1203#	1204#	1226#	1227#	1228#	1232#	1233#
	1249#	1269#	1273#	1276#	1281#	1282#	1294#	1305#	1306#	1310#	1311#	1323#	1332#	1333#	1337#
	1338#	1349#	1355#	1356#	1360#	1361#	1375#	1382#	1383#	1387#	1388#	1400#	1412#	1413#	1425#
	1426#	1430#	1431#	1443#	1453#	1454#	1462#	1463#	1467#	1468#	1479#	1486#	1487#	1495#	1496#
	1500#	1501#	1515#	1523#	1524#	1533#	1534#	1538#	1539#	1563#	1564#	1569#	1570#	1577#	1578#
	1604#	1605#	1610#	1611#	1618#	1619#	1645#	1646#	1651#	1652#	1659#	1660#	1687#	1688#	1693#
	1694#	1699#	1700#	1707#	1708#	1734#	1735#	1739#	1740#	1743#	1744#	1777#	1778#	1785#	1786#
	1787#	1796#	1797#	1798#	1812#	1821#	1822#	1823#	1839#	1840#	1854#	1855#	1859#	1860#	1861#
	1890#	1891#	1898#	1911#	1918#	1919#	1920#	1921#	1949#	1956#	1957#	1978#	1989#	1993#	2004#

CVRLBAO -- RLV12 DISKLESS
CVRLBA.P11 08-FEB-82 09:59

MACY11 30G(1063) 08-FEB-82 10:07 D 12
PAGE 49-5
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0146

POINTE	41														
PRINTB	2081	2177	3101	3104	3107	3110	3113	3116	3119	3122	3126	3133	3142	3173	3174
	3175	3176													
PRINTF	612	624	1104	1105	1106	2845	3012	3051	3188						
PRINTX	2072	2168	3149	3150	3157	3158	3161								
READER	465	467	469	471	473										
SETPRI	603	1249	1812	1822	1840	1860	1911	1920	2140	2179					
SFERR	199#	586	1135	1157	1179	1201	1226								
STARS	1120	1125	1142	1147	1164	1169	1186	1191	1208	1211	1237	1247	1286	1291	1315
	1320	1342	1346	1365	1370	1392	1397	1435	1440	1472	1476	1505	1510	1543	1551
	1582	1588	1623	1629	1664	1670	1712	1717	1748	1754	1802	1809	1827	1834	1865
	1870	1903	1907	1925	1935	1961	1973	2094	2101	2186	2195	2232	2245	2358	2366
	2424	2427	2461	2463	2492	2494	2539	2545	2668	2670					
SVC	16#	17													
XFER	59	64	1107#	1787#	2015#	2029#	2129#	2201#	2253#	2595#	2720#				
XFERF	57	101													
XFERT	104														

. ABS. 040362 000 CON RO REL GBL D

ERRORS DETECTED: 0

CVRLBA.BIC CVRLBA.LST/LI:TOC/CRF=SVC33/ML,CVRLBA.P11

RUN-TIME: 36 37 4 SECONDS

RUN-TIME RATIO: 327/78=4.1

CORE USED: 28K (55 PAGES)