

RL11,RLV11, RL11/RLV11/CTLR TS 1
RL01,RL02 CZRLGCO

AH-F110C-MC
FICHE 1 OF 1

JUL 1982
COPYRIGHT © 79-82
MADE IN USA



The main body of the document is a large grid of approximately 15 columns and 25 rows of data. Each cell in the grid contains a small, dense table or list of information, possibly representing a detailed schedule or a set of technical specifications. The text within these cells is extremely small and difficult to read, but the overall structure is a regular, repeating pattern of data blocks.

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F111C-MC
PRODUCT NAME: CZRLGCO RL11/RLV11 CONTROLLER TEST 1
DATE CREATED: 5-JAN-79
REVISED: 22-FEB-82
MAINTAINER: DIAGNOSTIC ENGINEERING - COLORADO
AUTHORS: D. CLAFLIN

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) 1979,1982 DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
1.1.1	STRUCTURE OF PROGRAM
1.1.2	DIAGNOSTIC INFORMATION
1.2	SYSTEM REQUIREMENTS
1.2.1	HARDWARE REQUIREMENTS
1.2.2	SOFTWARE REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	HOW TO RUN THIS DIAGNOSTIC
2.1.1	THE FIVE STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	CHAIN MODE OPERATION
2.3	DETAILS OF COMMANDS AND SYNTAX
2.3.1	TABLE OF COMMAND VALIDITY
2.3.2	COMMAND SYNTAX
2.4	EXTENDED P-TABLE DIALOGUE
2.5	HARDWARE PARAMETERS
2.6	SOFTWARE PARAMETERS
3.0	ERROR INFORMATION
3.1	ERROR REPORTING
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

1.1.1 STRUCTURE OF PROGRAM

THIS DIAGNOSTIC IS COMPATIBLE WITH BOTH XXDP+ AND ACT. IT CAN BE RUN STANDALONE UNDER XXDP+, AND CAN BE CHAINED UNDER XXDP+, ACT AND APT IN ACT MODE (SEE 2.2 "CHAIN MODE OPERATION" FOR DETAILS OF CHAINING PROCEDURE). IT IS A SINGLE PROGRAM FROM THE STANDPOINT OF THE DIAGNOSTIC USER, WHICH AT RUN TIME IS APPENDED TO A COMMON FRONT-END PIECE OF SUPERVISOR SOFTWARE THROUGH WHICH THE DIAGNOSTIC INTERFACES TO THE ENVIRONMENT AS IT EXECUTES. USING THE DEFAULT VALUES IN THE P TABLES, PROGRAM EXECUTES ONE PASS IN 11 SECONDS.

WHEN THIS DIAGNOSTIC IS STARTED, CONTROL GOES FIRST TO THE SUPERVISOR PORTION, WHICH WILL ASK CERTAIN "HARD CORE" QUESTIONS ABOUT THE ENVIRONMENT. THEN IT WILL ENTER COMMAND MODE, INDICATED BY A PROMPT CHARACTER (DR>). AT COMMAND MODE THE OPERATOR MAY ENTER ANY OF SEVERAL COMMANDS AS DESCRIBED IN 2.0 "OPERATING INSTRUCTIONS".

THE DIAGNOSTIC PROGRAM IS LOADED IN THE LOWER 8K OF MEMORY. THE DIAGNOSTIC SUPERVISOR CODING OCCUPIES 6.25K OF THE UPPER PART OF MEMORY JUST BELOW THE XXDP+ MONITOR WHICH RESIDES IN THE UPPERMOST 1.5K OF MEMORY SPACE.

1.1.2 DIAGNOSTIC INFORMATION

THE RL11/RLV11 CONTROLLER TEST (PART 1) IS A PDP-11 (LSI-11) BASED PROGRAM THAT WILL TEST THE CONTROLLER. IT STARTS BY TESTING BASIC INTERFACE LOGIC, REGISTER MANIPULATION AND FUNCTIONALITY WHICH INCLUDES NOOP, GET STATUS, READ HEADERS AND SEEK OPERATIONS. IT IS AIMED AT FULLY TESTING THE CONTROLLER IN THESE AREAS, BUT BY DEFAULT ALSO EXERCISES THE DRIVE.

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE REQUIREMENTS

- * PDP-11/LSI-11 PROCESSOR WITH 16K OR MORE OF MEMORY
- * CONSOLE DEVICE (LA30,LA36,VT50,ETC.)
- * 1 OR 2 RL11/RLV11 CONTROLLER(S) WITH:
 - 1 - 8 RL01 DRIVES WITH RL01K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
 - 1 - 8 RL02 DRIVES WITH RL02K CARTRIDGES CONTAINING A 'BAD SECTOR FILE'
- * LINE PRINTER (OPTIONAL)

1.2.2 SOFTWARE REQUIREMENTS

CZRLGB0 RL11/RLV11 CTLR TST 1
(FORMERLY CZRLAB)

1.3 RELATED DOCUMENTS AND STANDARDS

RL01 DISK SUBSYSTEM USER'S GUIDE (EK-RL01-UG-002)
XXDP+/SUPERVISOR USER'S MANUAL

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THE RL01/02 SUBSYSTEM SHOULD HAVE SUCCESSFULLY RUN THE FOLLOWING PROGRAMS:

CVRLAB0 RLV11 RL01 DISKLESS TEST (RLV11 ONLY)

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE RL01/02 SUBSYSTEM IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE FIVE STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP+ PROCEDURES. START THE EXECUTION OF THE XXDP+ MONITOR BY USING THE APPROPRIATE BOOTSTRAP PROGRAM. THE MONITOR WILL PRINT A MESSAGE IDENTIFYING ITSELF AND REQUESTING THAT THE CURRENT DATE BE ENTERED. AN EXAMPLE OF THIS MESSAGE IS GIVEN BELOW FOR THE XXDP+ MONITOR.

```
CHMDKAO XXDP+ DK MONITOR NNK
BOOTED VIA UNIT#: 0
ENTER DATE (DD-MMM-YY):
```

AFTER THE DATE HAS BEEN ACCEPTED BY THE MONITOR. THE RESTART ADDRESS OF THE MONITOR IS PRINTED. THEN THE FOLLOWING TWO QUESTIONS ARE ASKED:

```
50 HZ? N
LSI?   N
```

THE DEFAULTS ARE BOTH 'NO'. TYPE 'R' AND THE PROGRAM NAME TO RUN THE PROGRAM. DO NOT TYPE THE EXTENSION.

WHEN THIS DIAGNOSTIC IS STARTED, THE FOLLOWING STEPS WILL OCCUR:

```
*****
* STEP 1 *
*****
```

THE DIAGNOSTIC WILL ISSUE THE PROMPT 'DR>'. FROM THIS POINT UNTIL THE TIME WHEN YOU RESTART XXDP+, YOU WILL BE TALKING TO THE DIAGNOSTIC, NOT XXDP+. WE WILL REFER TO THE PRESENCE OF THIS PROMPT AS BEING IN DIAGNOSTIC COMMAND MODE, AS OPPOSED TO XXDP+ COMMAND MODE.

AT THIS POINT YOU WILL ENTER A "START" COMMAND. THIS IS NOT THE SAME AS THE XXDP+ "START" COMMAND, WHICH YOU ALREADY ISSUED IN RESPONSE TO THE XXDP+ DOT PROMPT. THIS "START" COMMAND CAN TAKE A NUMBER OF SWITCHES AND FLAGS (ALL OPTIONAL) AND THE DETAILS OF THESE ARE SET FORTH IN 2.3 "DETAILS OF COMMANDS AND SYNTAX". HOWEVER, IN ORDER TO USE THE PROGRAM, ALL YOU NEED TO SAY IS SOMETHING LIKE THIS:

```
STA/PASS:1/FLAGS:HOE
```


THINGS TO NOTE HERE:

1. ONLY THE FIRST THREE CHARACTERS OF THIS OR ANY COMMAND AT THE "DR>" LEVEL NEED TO BE TYPED.
2. THE "PASS" SWITCH SPECIFIES HOW MANY PASSES YOU DESIRE. A PASS CONSISTS OF RUNNING THE FULL DIAGNOSTIC AGAINST ALL UNITS BEING TESTED (THIS WILL BE EXPLAINED SHORTLY). ONE PASS IS SPECIFIED IN THE ABOVE EXAMPLE.
3. THE "FLAGS" SWITCH MAY SPECIFY ANY OF A NUMBER OF FLAGS, BUT THE MAIN USEFUL ONES ARE:

PNT	PRINT NUMBER OF TEST BEING EXECUTED
LOE	LOOP ON ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

THE HOE FLAG IS SPECIFIED IN THE ABOVE EXAMPLE (WE'LL SEE WHY SHORTLY).

 * STEP 2 *

WHEN YOU HAVE TYPED IN A "START" COMMAND, THE DIAGNOSTIC WILL COME BACK WITH THE QUESTION "# UNITS?" TO WHICH YOU SHOULD RESPOND BY TYPING IN THE NUMBER OF DEVICES YOU WISH TO TEST.

A WORD OF WARNING HERE: THE NUMBER OF UNITS DEPENDS ON THE TARGET DEVICE OF THE DIAGNOSTIC. FOR EXAMPLE, IF THE DIAGNOSTIC IS DIRECTED AT A DISK DRIVE, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF DRIVES TO BE TESTED. WHEREAS IF THE DIAGNOSTIC WAS DIRECTED AT THE DISK CONTROLLER, THEN THE NUMBER OF UNITS WOULD BE THE NUMBER OF CONTROLLERS. THE TARGET DEVICE OF A DIAGNOSTIC CAN ALWAYS BE DETERMINED BY INSPECTING THE "HEADER" STATEMENT NEAR THE BEGINNING OF THE SOURCE CODE. ONE OF THE OPERANDS OF THIS "HEADER" STATEMENT SHOULD BE THE DEVICE TYPE OF THE DIAGNOSTIC.

 * STEP 3 *

WHEN YOU HAVE TYPED IN THE NUMBER OF UNITS TO BE TESTED, THE DIAGNOSTIC WILL ASK YOU THE "HARDWARE QUESTIONS". THE ANSWERS TO THESE QUESTIONS ARE USED TO BUILD TABLES IN CORE, CALLED "HARDWARE P-TABLES". ONE HARDWARE P-TABLE WILL BE BUILT FOR EACH UNIT TO BE TESTED.

THERE ARE SEVERAL HARDWARE QUESTIONS AND THE ENTIRE SERIES WILL BE POSED N TIMES, WHERE N IS THE NUMBER OF UNITS.

THIS REPRESENTS A NEW PHILOSOPHY IN DIAGNOSTIC ENGINEERING. DIAGNOSTICS IN THE FUTURE WILL NOT BE WRITTEN TO AUTOSIZE OR ASSUME STANDARD ADDRESSES: INSTEAD, THEY WILL ASK THE OPERATOR FOR ALL THE INFORMATION THEY NEED TO TEST THE DEVICE.

* STEP 4 *

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS (SEC 2.5) FOR ALL THE UNITS, YOU WILL BE ASKED "CHANGE SW?" IF YOU WANT TO BE ASKED THE SOFTWARE QUESTIONS THAT DETERMINE THE BEHAVIOR OF THIS PROGRAM, TYPE "Y". IF YOU WANT TO TAKE ALL THE DEFAULTS TO THESE QUESTIONS, TYPE "N". IF YOU TYPE "Y" YOU WILL BE ASKED THE SOFTWARE QUESTIONS (SEC 2.6), AND THE ANSWERS WILL BE PUT INTO THE SOFTWARE P-TABLE IN THE PROGRAM. THE SERIES OF QUESTIONS WILL BE ASKED JUST ONCE, REGARDLESS OF THE NUMBER OF UNITS TO BE TESTED.

* STEP 5 *

AFTER YOU HAVE ANSWERED THE SOFTWARE QUESTIONS, THE DIAGNOSTIC WILL BEGIN TO EXECUTE THE HARDWARE TEST CODE. THERE ARE SEVERAL THINGS THAT CAN HAPPEN NEXT, DEPENDING ON WHETHER A HARDWARE ERROR IS ENCOUNTERED AND ALSO ON WHAT SWITCH VALUES YOU SELECTED ON THE START COMMAND. CONSIDER THE POSSIBILITIES:

1. IF NO ERROR IS ENCOUNTERED, THEN THE DIAGNOSTIC WILL SIMPLY EXECUTE THE DESIRED NUMBER OF PASSES AND RETURN TO COMMAND MODE (PROMPT DR>).
2. IF AN ERROR IS ENCOUNTERED, THEN ONE OF THREE THINGS HAPPENS, DEPENDING ON THE SETTINGS OF THE HOE AND LOE FLAGS.

HOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND THE DIAGNOSTIC WILL RETURN TO COMMAND MODE.

LOE SET: THE DIAGNOSTIC WILL LOOP ENDLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.

NEITHER HOE NOR LOE SET: THE ERROR WILL BE REPORTED ON THE CONSOLE AND NORMAL EXECUTION WILL RESUME AS IF NO ERROR HAD OCCURRED.

2.1.2 SAMPLE RUN-THROUGH

LET'S SEE HOW ALL THIS WORKS IN A REAL SITUATION. RECALL THAT WE ENTERED THE COMMAND "STA/PASS:1/FLAGS:HOE". THIS WOULD BE A VERY TYPICAL WAY TO RUN THE DIAGNOSTIC. IF NO ERRORS ARE ENCOUNTERED, THE SINGLE REQUESTED PASS WILL BE EXECUTED AND THE PROMPT WILL BE RE-ISSUED.

IF AN ERROR IS ENCOUNTERED, THE ERROR WILL BE REPORTED AND THE PROMPT WILL BE REISSUED (BECAUSE THE HOE FLAG IS SET). AT THIS POINT THERE ARE FOUR DIFFERENT WAYS YOU CAN GET THE PROGRAM GOING AGAIN:

1. ISSUE ANOTHER "START" COMMAND (THUS GOING THRU ALL OF STEPS 1, 2, 3, 4, AND 5 AGAIN).
2. ISSUE A "RESTART" COMMAND (SAME AS START COMMAND EXCEPT THAT THE HARDWARE QUESTIONS ARE NOT ASKED)
3. ISSUE A "CONTINUE" COMMAND (EXECUTION WILL RESUME AT THE BEGINNING OF THE PARTICULAR HARDWARE TEST (MOST DIAGNOSTICS CONSIST OF A NUMBER OF THESE) THAT IT WAS IN WHEN THE ERROR HALT OCCURRED. NO QUESTIONS ASKED.
4. ISSUE A "PROCEED" COMMAND: EXECUTION WILL RESUME AT THE INSTRUCTION FOLLOWING THE ERROR REPORT (THIS IS A SPECIAL COMMAND AND CAN BE ISSUED ONLY AT A HALT

THE MOST TYPICAL THING TO DO HERE IS TO ISSUE THE PROCEED, BUT WITH DIFFERENT FLAG SETTINGS. PROBABLY YOU WOULD WANT TO SAY:

```
PRO/FLAGS:IER:LOE:HOE=0
```

THIS WILL DO THE FOLLOWING:

1. TURN ON THE IER (INHIBIT ERROR PRINTOUT) FLAG
2. TURN ON THE LOE FLAG
3. TURN OFF THE HOE FLAG
4. RESUME EXECUTION AT INSTRUCTION AFTER ERROR REPORT

THE DIAGNOSTIC WILL NOW LOOP ON THE BLOCK OF CODE THAT DETECTED AND REPORTED THE ERROR, BUT NO ERROR PRINTOUT WILL OCCUR. THUS YOU CAN STUDY THE ERROR OR SCOPE IT OR WHATEVER.

WHEN YOU'VE SEEN ENOUGH, YOU MAY HIT CONTROL/C. THIS WILL TAKE YOU OUT OF THE LOOP AND PUT YOU BACK INTO COMMAND MODE. YOU NOW HAVE THREE CHOICES:

1. START
2. RESTART
3. CONTINUE

LET'S SAY YOU'VE REPAIRED THE DEFECT FOUND ABOVE AND WANT TO FINISH RUNNING THE DIAGNOSTIC. YOU WOULD TYPE

CON/FLAGS:HOE:IER=0:LOE=0

THIS WILL RESTORE THE FLAGS TO THEIR ORIGINAL VALUES AND RESUME EXECUTION AT THE BEGINNING OF THE HARDWARE TEST YOU WERE IN. IF THE ERROR DOES NOT RECUR, THE EXECUTION WILL FLOW RIGHT ON THRU TO THE NEXT ERROR OR TO END OF PASS.

IF AT END OF PASS YOU WANT TO RUN THE DIAGNOSTIC AGAIN, YOU HAVE TWO CHOICES:

1. START
2. RESTART

YOU WOULD CHOOSE ONE, DEPENDING ON WHETHER YOU WANTED TO ANSWER THE HARDWARE QUESTIONS AGAIN.

THE FULL PRINT-OUT FROM THE ABOVE DIALOGUE MIGHT LOOK LIKE THIS
(O=OPERATOR, D=DIAGNOSTIC):

	BY WHOM ENTERED: -----
.R CZRLGB	O
DRS LOADED	D
DIAG. RUN-TIME SERVICES REV D APR-79	D
CZRLG-B-0	D
CZRLG TESTS CONTROLLER FUNCTIONS, INTERFACE LOGIC, REGISTER OPERATION	D
UNIT IS RL01, RL02	D
DR>STA/PASS:1/FLAGS:HOE	D.O
# UNITS (D) ? 2	D.O
UNIT 0	D
RL11 (L) Y ?	D.O
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
BR LEVEL (O) 5 ?	D.O
DRIVE TYPE = RL01 (L) Y ?	D.O (N=RL02)
DRIVE (O) 0 ?	D.O
UNIT 1	D
RL11 (L) Y ?	D.O
BUS ADDRESS (O) 174400 ?	D.O
VECTOR (O) 160 ?	D.O
BR LEVEL (O) 5 ?	D.O
DRIVE TYPE = RL01 (L) ? Y	D.O (N=RL02)
DRIVE (O) 0 ? 1	D.O
CHANGE SW (L) ? Y	D.O
DROP ON ERROR LIMIT (L) N ?	D.O
CZRLG HRD ERR 00004 TST 003 SUB 002 PC:004130	
ERR HLT	
DR>PRO/FLAGS:IER:LOE:HOE=0	D.O

 AT THIS POINT THE DIAGNOSTIC IS LOOPING ON THE
 ERROR WITHOUT PRINTING ANYTHING. YOU CAN SCOPE
 THE ERROR UNTIL YOU HAVE LOCATED IT, THEN ^C OUT.
 TYPING ^C ABORTS THE FUNCTION IN PROGRESS AND
 RETURNS THE XXDP+ MONITOR TO COMMAND MODE.

```

^C                                0
DR>CON/FLAGS:HOE:IER:LOE=0       D.0
CHANGE SW (L) ? N                 D.0
CZRLG EOP 1                       D
^C
DR>RESTART/PASS:1                 D.0
CHANGE SW (L) ? N                 D.0
-----
-----
-----
-----

```

2.2 CHAIN MODE OPERATION

CHAIN MODE OPERATION CONSISTS OF THE SEQUENTIAL EXECUTION OF PROGRAMS WITHOUT OPERATOR INTERVENTION. ONLY PROGRAMS THAT HAVE BEEN MODIFIED TO RUN IN CHAIN MODE CAN BE CHAINED. CHAINABLE PROGRAMS ARE IDENTIFIED IN THE DIRECTORY BY A BIC EXTENSION.

TO RUN CHAIN MODE, THE XXDP+ MONITOR USES AN ASCII FILE (KNOWN AS A CHAIN FILE) LISTING THE PROGRAMS TO BE RUN AND THE NUMBER OF PASSES EACH PROGRAM SHOULD RUN. THIS FILE MUST BE ON THE SYSTEM DEVICE.

A CHAIN FILE MAY BE GENERATED BY USE OF THE XTECO TEXT EDITOR. THE FILE MUST HAVE A CCC EXTENSION. THE CHAIN FILE MAY CONTAIN ANY OF THE COMMANDS SUPPORTED BY THE XXDP+ MONITOR. THE COMMANDS IN THE ASCII FILE ARE EXECUTED IN THE ORDER IN WHICH THEY ARE ENCOUNTERED. COMMENTS MAY BE INCLUDED IN THE FILE.

TO EXECUTE A CHAIN FILE THE USER TYPES:

C FILNAM <CR> OR

C FILNAM/QV<CR>

IN THE FIRST CASE THE PASS COUNT SPECIFIED IN THE CHAIN FILE IS USED BY THE XXDP+ MONITOR TO DETERMINE THE NUMBER OF PASSES TO EXECUTE EACH PROGRAM. IN THE SECOND CASE THE PASS COUNT IS NOT USED AND EACH PROGRAM IS EXECUTED ONLY ONCE. THE /QV SWITCH PROVIDES A SINGLE EXECUTION MODE OF OPERATION OF QUICK VERIFY.

WHEN PROGRAMS ARE RUN IN CHAIN MODE, THE SOFTWARE SWITCH REGISTER SHOULD BE SET TO 000000. THE XXDP+ MONITOR PRINTS EACH COMMAND TAKEN FROM THE CHAIN FILE AND THEN EXECUTES THE COMMAND. WHEN THE LAST COMMAND- OTHER THAN ANOTHER C COMMAND HAS BEEN EXECUTED THE XXDP+ MONITOR TERMINATES CHAIN MODE AND TYPES A PROMPT (.). IT IS READY TO ACCEPT ANOTHER COMMAND FROM THE CONSOLE. IF THE LAST COMMAND IS ANOTHER C COMMAND, THE CHAIN MODE WILL CONTINUE AND THE CHAIN FILE SPECIFIED BY THIS NEW C COMMAND WILL BE USED. IF THE USER WISHES TO TERMINATE CHAIN MODE BEFORE ITS NORMAL TERINATION HE MAY DO SO BY TYPING A CNTL-C. HOWEVER, THE MONITOR WILL NOT ABORT THE CHAIN MODE UNTIL IT RECEIVES PROGRAM CONTROL FROM THE PROGRAM CURRENTLY RUNNING.

2.3 DETAILS OF COMMANDS AND SYNTAX

2.3.1 TABLE OF COMMAND VALIDITY

THERE ARE FOUR WAYS OF ENTERING DIAGNOSTIC COMMAND MODE, AND DIFFERENT SUBSETS OF THE DIAG COMMAND SET ARE AVAILABLE WITH EACH:

<u>HOW ENTERED</u>	<u>LEGAL COMMANDS</u>
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS EXIT
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSES	START RESTART PRINT DISPLAY FLAGS ZFLAGS EXIT
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS EXIT

4. AN ERROR WAS ENCOUNTERED
WITH THE HOE FLAG SET

START
RESTART
CONTINUE
PROCEED
PRINT
DISPLAY
FLAGS
ZFLAGS
EXIT

2.3.2 COMMAND SYNTAX

STA(RT)/TESTS:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. THE MESSAGE "# UNITS?" IS PRINTED. THE START COMMAND MAY BE ISSUED WHEN DIAGNOSTIC COMMAND MODE HAS BEEN ENTERED VIA ONE OF THE FOLLOWING: A) OPERATOR TYPED 'RUN DIAGNOSTIC' B) DIAGNOSTIC FINISHED EXECUTING C) ERROR WAS ENCOUNTERED WITH HOE FLAG SET D) OPERATOR ENTERED CONTROL/C. AFTER THE OPERATOR RESPONDS TO "# UNITS?", THE HARDWARE DIALOGUE IS INITIATED. WHEN IT IS COMPLETED, THE QUESTIONS "CHANGE SW?" IS ISSUED, AND THE ANSWERS, IF GIVEN, BECOME THE NEW DEFAULTS. THEREFORE IT IS NECESSARY TO RELOAD THE PROGRAM IN ORDER TO RETURN TO THE LOAD DEFAULTS.

THE SWITCH ARGUMENTS ARE AS FOLLOWS:

"TEST-LIST" IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS.

"PASS-CNT" IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING TEST EXECUTION. "FLAG-LIST" IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED

LOE LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUB-TEST, OR TEST) CONTAINING THE ERROR

IER INHIBIT ERROR REPORTING
IBE INHIBIT BASIC ERROR REPORTS
IXE INHIBIT EXTENDED ERROR REPORTS
PRI DIRECT ALL MESSAGES TO A LINE PRINTER
PNT PRINT NUMBER OF TEST BEING EXECUTED
BOE BELL ON ERROR
UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR INHIBIT STATISTICAL REPORTS
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
ADR EXECUTE AUTODROP CODE
LOT LOOP ON TEST
EVL EVALUATE

THESE FLAGS REPLACE THE USE OF THE HARDWARE SWITCH REGISTER. UNDER THE SUPERVISOR THERE IS NO ACCESS TO THE HARDWARE SWITCH REGISTER.

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.

'EOP-INCR' IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS.

RES(TART)/TEST:TEST-LIST/PASS:PASS-CNT/FLAGS:FLAG-LIST/EOP:EOP-INCR/
UNITS:UNIT-LIST

THE DIAGNOSTIC IN CORE IS EXECUTED IN ACCORDANCE WITH THE SWITCHES SPECIFIED. HOWEVER, NEW 'P-TABLES' ARE NOT BUILT. INSTEAD, THE ONES IN CORE ARE USED.

THE QUESTION "CHANGE SW?" IS ASKED AND THE ANSWERS GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMAND MODE HAS BEEN ENTERED VIA A) DIAGNOSTIC IS FINISHED B) HALT ON ERROR C) CONTROL/C.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. 'UNIT-LIST' IS A SEQUENCE OF LOGICAL UNIT NUMBERS RANGING FROM 1 THRU N (N = NUMBER OF UNITS BEING TESTED) SPECIFYING WHICH UNITS ARE TO BE TESTED. THE LOGICAL UNIT NUMBER DESIGNATES THE POSITION OF THE P-TABLE IN CORE, ACCORDING TO THE ORDER IN WHICH THEY WERE BUILT. THE UNITS SPECIFIED MUST NOT HAVE BEEN DROPPED BY THE OPERATOR DROP COMMAND. THE UNIT-LIST DEFAULTS TO 'ALL THAT HAVE NOT BEEN DROPPED BY OPERATOR COMMAND'. THE EFFECT OF THE UNIT-LIST LASTS UNTIL THE NEXT START (WHERE IT IS AUTOMATICALLY RESET TO 'ALL') OR THE NEXT RESTART.
2. ALL UNSPECIFIED FLAG SETTINGS ARE UNCHANGED.

CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>

COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE RE-EXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

1. DEFAULT FOR PASS-CNT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART
2. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

PRO(CCEED)/FLAGS:<FLAG-LIST>

COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

THE SWITCH ARGUMENTS ARE THE SAME AS THE START COMMAND EXCEPT:

1. UNSPECIFIED FLAG SETTINGS ARE UNCHANGED

EXIT

RETURN TO XXDP+ PROMPT MODE.

DRO(P)/UNITS:UNIT-LIST

THE UNITS SPECIFIED ARE DROPPED FROM TESTING UNTIL THEY ARE ADDED BACK OR UNTIL A START COMMAND IS GIVEN. A DROP CANNOT BE FOLLOWED BY A PROCEED.

THERE IS ALSO A 'DROP' MACRO INTERNAL TO THE DIAGNOSTIC, WHICH GIVES THE FACILITY OF AUTO-DROPPING. THE DURATION OF A PROGRAM DROP, HOWEVER, IS ONLY UNTIL THE NEXT START OR RESTART.

ADD/UNITS:UNIT-LIST

THE UNITS SPECIFIED ARE ADDED BACK (THEY MUST HAVE BEEN PREVIOUSLY DROPPED BY THE DROP COMMAND) TO THE TEST SEQUENCE. AN ADD CANNOT BE FOLLOWED BY A PROCEED.

PRI(NT)

ALL STATISTICS TABLES ACCUMULATED BY THE DIAGNOSTIC ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

DIS(PLAY)/UNITS:<UNIT-LIST>

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO DESIGNATED.

FLA(GS)

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

ZFL(AGS)

ALL FLAGS ARE CLEARED.

2.4 EXTENDED P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N), SPACE IN CORE IS ALLOCATED FOR 'N' P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO-ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 8 RL UNITS, AND THAT THERE ARE FIVE (5) HARDWARE PARAMETERS FOR EACH (5 SLOTS IN THE P-TABLE, 5 HARDWARE QUESTIONS IN THE DIALOGUE).

FOLLOWING IS THE DIALOGUE FOR THIS 8 RLOX DRIVE SYSTEM. THIS SYSTEM HAS TWO (2) RL11 TYPE CONTROLLERS ALL TO BE SET AT 'BR LEVEL' 5. THE FIRST 4 DRIVES ARE RLO1'S AND THE LAST 4 DRIVES ARE RLO2'S (ON THE SECOND CONTROLLER):

UNITS (D) ? 8

UNIT 0

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ?

VECTOR (O) 160 ?

BR LEVEL (O) 5 ?

DRIVE TYPE = RLO1 (L) Y ?

DRIVE (O) ? 0-3

UNIT 4

RL11 (L) Y ?

BUS ADDRESS (O) 174400 ? 175400

VECTOR (O) 160 ? 164

BR LEVEL (O) 5 ?

DRIVE TYPE = RLO1 (L) Y ? N

DRIVE (O) ? 0-3

THE FIRST TIME THRU THE P-TABLE QUESTIONS THE DEFAULT VALUES ARE USED FOR THE CONTROLLER TYPE (QUESTION #1), CSR ADDRESS OF THE CONTROLLER (QUESTION #2), THE CONTROLLER VECTOR ASSIGNMENT (QUESTION #3), THE DRIVE TYPE (QUESTION #5), AND THE "BR LEVEL" (QUESTION #4). THE ACTUAL UNIT NUMBERS OF THE RL01'S FOR QUESTION #6 WAS ASSIGNED 0 THRU 3 FOR THE FIRST 4 P-TABLE SLOTS.

THE SECOND TIME THRU THE P-TABLE QUESTIONS (FOR THE RL02 ASSIGNMENT ON THE SECOND CONTROLLER), THE FIRST QUESTION DEFAULTED TO "RL11" TYPE CONTROLLER. THE SECOND QUESTION WAS ANSWERED TO REFLECT THE CHANGE IN CSR ADDRESS FOR THE RL02 CONTROLLER (175400). THE SECOND CONTROLLER'S VECTOR WAS ALSO CHANGED TO 164 IN QUESTION #3. THE RL02 TEST UNIT NUMBERS WERE ASSIGNED VALUES 0 TO 3 IN QUESTION #6 AND THE DRIVE TYPE WAS SET FOR RL02'S FOR THE REMAINING 4 UNITS IN QUESTION #5. QUESTION #4 WAS DEFAULTED USING THE "BR LEVEL" FROM THE FIRST PASS.

2.5 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

RL11 (L) Y?

ANSWER YES(Y) IF YOU HAVE AN RL11 CONTROLLER, NO(N) IF YOU HAVE AN RLV11 CONTROLLER.

BUS ADDRESS (0) 174400?

ANSWER WITH THE BUS ADDRESS OF THE CONTROLLER.

VECTOR (0) 160?

ANSWER WITH THE INTERRUPT VECTOR OF THE CONTROLLER.

BR LEVEL (0) 5?

ANSWER WITH THE INTERRUPT PRIORITY OF THE CONTROLLER.

DRIVE TYPE = RL01 (L) ?

ANSWER NO (N) IF DRIVE IS AN RL02

DRIVE (0) 0?

ANSWER WITH THE DRIVE(S) CONNECTED TO THE CONTROLLER.

2.6 SOFTWARE PARAMETERS

THE FOLLOWING QUESTIONS ARE ASKED IF REQUESTED ON A START, RESTART, OR CONTINUE. THEY ALLOW FLEXIBILITY IN THE WAY THE PROGRAM BEHAVES. THE SOFTWARE PARAMETERS GIVE THE PROGRAM FLEXIBILITY IN THE WAY IT RUNS. THE PARAMETERS CAN BE MODIFIED ON A START, RESTART, OR CONTINUE BY ANSWERING (Y)ES TO THE FOLLOWING QUESTION:

"CHANGE SW ?"

A YES ANSWER WILL ASK THE FOLLOWING SOFTWARE PARAMETER QUESTION, WITH THE PRESENT DEFAULT VALUE PRINTED TO THE LEFT OF THE QUESTION MARK. (THE LAST ANSWER GIVEN IS THE DEFAULT) THE DEFAULT IS TAKEN ON A <CR>.

"DROP ON ERROR LIMIT (L) Y?"

TO ALLOW THE UNIT TO BE DROPPED ONCE A PREDETERMINED NUMBER OF ERRORS ARE ENCOUNTERED.

3.0 ERROR INFORMATION

ALL ERRORS ARE PRINTED VIA CONSOLE DEVICE. THE ERROR INCLUDES ERROR NUMBER, TYPE AND PROGRAM LOCATION. ERRORS INCLUDE REGISTERS BEFORE AND AT ERROR WITH RELEVANT DATA.

3.1 ERROR REPORTING

ALL ERROR INFORMATION IS PRINTED ON THE CONSOLE DEVICE. ERROR REPORTS ARE AIMED AT BEING SELF EXPLANATORY. THE GENERAL FORMAT IS:

DZRL? XXX ERR YYYYY TST ZZZ SUB PPP PC: RRRRRR

WHERE:

? IS PROGRAM LETTER
XXX IS SFT - SOFT ERROR
HRD - HARD ERROR
DV FAT - DEVICE FATAL ERROR
SYS FAT - SYSTEM FATAL ERROR
YYYYY IS THE ERROR NUMBER
ZZZ IS THE TEST NUMBER
PPP IS THE SUBTEST NUMBER
RRRRRR IS THE PROGRAM LISTING LOCATION

ERRORS GIVE THE REGISTER CONTENTS BEFORE AND AFTER THE ERROR ALONG WITH A ONE LINE DESCRIPTION AND RELEVANT DATA.

EXAMPLE:

ONE LINE DESCRIPTION
(OPTIONAL SECOND LINE)
(OPTIONAL THIRD LINE)
BEFORE COMMAND: CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX
TIME OF ERROR: CS:XXXXXX BA:XXXXXX DA:XXXXXX MP:XXXXXX XXXXXX
XXXXXX

3.2 ERROR HALTS

ERROR HALTS ARE SUPPORTED PER DESCRIBED IN THE PREVIOUS SECTION WITH /FLAG:HOE. THERE ARE NO OTHER HALTS.

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THIS PROGRAM WILL NOT GIVE ANY PERFORMANCE REPORTS.

4.2 PROGRESS REPORTS

THIS PROGRAM WILL NOT GIVE ANY PROGRESS REPORTS.

5.0 DEVICE INFORMATION TABLES

THE RL11/RLV11 CONTROLLER HAS THE FOLLOWING FOUR(4) REGISTERS FOR CONTROL OF THE SUBSYSTEM.

RLCS - CONTROL AND STATUS REGISTER (XXXXX0)

- BIT 15 - COMPOSITE ERROR
- BIT 14 - DRIVE ERROR
- BIT 13 - NON EXISTANT MEMORY ERROR
- BIT 12 - HEADER NOT FOUND (WITH BIT 10 SET)
- DATA LATE (WITH BIT 10 CLEAR)
- BIT 11 - HEADER CRC (WITH BIT 10 SET)
- DATA CRC (WITH BIT 10 CLEAR)
- BIT 10 - OPERATION INCOMPLETE
- BIT 9/8 - DRIVE SELECT (0-3)
- BIT 7 - CONTROLLER READY
- BIT 6 - INTERRUPT ENABLE
- BIT 5 - EXTENDED BUS ADDRESS (BIT 17)
- BIT 4 - EXTENDED BUS ADDRESS (BIT 16)

BIT 3-1 - FUNCTION CODE

- 0 - NOP (PDP-11) MAINT (LSI-11)
- 1 - WRITE CHECK
- 2 - GET DRIVE STATUS
- 3 - SEEK
- 4 - READ HEADER
- 5 - WRITE DATA
- 6 - READ DATA
- 7 - READ WITHOUT HEADER COMPARE

BIT 0 - DRIVE READY

RLBA - BUS ADDRESS REGISTER (XXXXX2)

BITS 15-1 BUS ADDRESS OF DATA TRANSFER
BIT 0 SHOULD BE 0

RLDA - DISK ADDRESS REGISTER (XXXXX4)

FOR READ/WRITE FUNCTIONS

BIT 15-7 - CYLINDER ADDRESS FOR TRANSFER
BIT 6 - SURFACE FOR TRANSFER
BIT 5-0 - SECTOR FOR TRANSFER (1-40.)

FOR SEEK FUNCTION

BIT 15-7 - DIFFERENCE TO NEW CYLINDER
BIT 6-5 - MUST BE ZERO (0)
BIT 4 - SURFACE (0=UPPER, 1=LOWER)
BIT 3 - MUST BE ZERO (0)
BIT 2 - SEEK DIRECTION(1=IN / 0=OUT)
BIT 1 - MUST BE ZERO (0)
BIT 0 - MUST BE ONE (1)

FOR GET STATUS FUNCTION

BIT 15-4 - IGNORED SHOULD BE ZERO (0)
BIT 3 - DRIVE RESET
BIT 2 - MUST BE ZERO (0)
BIT 1 - MUST BE ONE (1)
BIT 0 - MUST BE ONE (1)

RLMP - MULTIPURPOSE REGISTER

FOR READ/WRITE FUNCTION

BIT 15 - 0 - WORD COUNT (TWO'S COMPLIMENT)

FOR READ HEADER FUNCTION

BIT 15-0 - DISK HEADER OF SECTOR (FIRST READ)
 - ZERO WORD (SECOND READ)
 - HEADER CRC (THIRD READ)

FOR GET STATUS FUNCTION

HAS DRIVE STATUS

BIT 15 - WRITE DATA ERROR
BIT 14 - CURRENT HEAD ERROR (CHE)
BIT 13 - WRITE LOCK STATUS (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 - DRIVE TYPE IS RL02 IF SET
BIT 6 - SURFACE (0=UPPPER, 1=LOWER)
BIT 5 - COVER OPEN
BIT 4 - HEADS HOME
BIT 3 - BRUSHES HOME
BIT 2-0 - STATE BITS
 0 - LOAD STATE
 1 - SPIN UP
 2 - BRUSH CYCLE
 3 - LOAD HEADS
 4 - SEEK - TRACK COUNTING
 5 - SEEK - LINEAR MODE
 6 - UNLOAD HEADS
 7 - SPIN DOWN

6.0 TEST SUMMARIES

TEST 1 - RLCS ADDRESSABILITY

THIS TEST WILL CHECK THAT THE CONTROL AND STATUS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

TEST 2 - RLBA ADDRESSABILITY

THIS TEST WILL CHECK THAT THE BUS ADDRESS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

TEST 3 - RLDA ADDRESSABILITY

THIS TEST WILL CHECK THAT THE DISK ADDRESS REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

TEST 4 - RLMP ADDRESSABILITY

THIS TEST WILL CHECK THAT THE MULTIPURPOSE REGISTER CAN BE ACCESSED. IF THE REGISTER CANNOT BE ACCESSED THE PROCESSOR WILL TRAP TO LOCATION 4, WHICH IS SET UP TO HANDLE THE TRAP.

TEST 5 - READ WRITE OF RLCS

THIS TEST WILL ATTEMPT TO WRITE RLCS BITS 9-1 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED. BIT 7 (CONTROLLER READY) IS ALWAYS WRITTEN AS A 1 SO NOT TO INITIATE A FUNCTION. BITS 15, 14 AND 0 ARE TREATED AS DON'T CARE FOR THIS TEST.

TEST 6 - READ WRITE OF RLBA

THIS TEST WILL ATTEMPT TO WRITE RLBA BITS 15-0 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED. BIT 0 ON A RL11 SHOULD ALWAYS COME BACK AS A 0, WHILE ON AN RLV11 IT IS LOADABLE.

TEST 7 - READ WRITE OF RLDA

THIS TEST WILL ATTEMPT TO WRITE RLDA BITS 15-0 AND READ THEM BACK. WALKING AND GROWING 0'S AND 1'S ARE USED.

TEST 8 - BIS OF RLCS

THIS TEST WILL USE THE 11 INSTRUCTION 'BIS' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLCS WORKS. BITS 9-1 ARE USED, BIT SETTING IN WALKING AND GROWING 0'S AND 1'S. BIT 7 (CONTROLLER READY) IS ALWAYS SET. BITS 15, 14 AND 1 ARE DON'T CARES.

TEST 9 - BIC OF RLCS

THIS TEST WILL USE THE 11 INSTRUCTION 'BIC' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLCS WORKS. BITS 9-1 ARE USED, BIT CLEARING IN WALKING AND GROWING 0'S AND 1'S. BIT 7 (CONTROLLER READY) IS ALWAYS SET. BITS 15, 14 AND 1 ARE DON'T CARES.

TEST 10 - BIS OF RLBA

THIS TEST WILL USE THE 11 INSTRUCTION 'BIS' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLBA WORKS. BITS 15-0 ARE BIT SET USING GROWING AND WALKING 0'S AND 1'S. BIT 0 CANNOT SET ON A RL11, BUT CAN ON A RLV11.

TEST 11 - BIC OF RLBA

THIS TEST WILL USE THE 11 INSTRUCTION 'BIC' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLBA WORKS. BITS 15-0 ARE BIT CLEARED USING GROWING AND WALKING 0'S AND 1'S.

TEST 12 - BIS OF RLDA

THIS TEST WILL USE THE 11 INSTRUCTION 'BIS' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLDA WORKS. BITS 15-0 ARE BIT SET USING GROWING AND WALKING 0'S AND 1'S.

TEST 13 - BIC OF RLDA

THIS TEST WILL USE THE 11 INSTRUCTION 'BIC' TO SHOW THAT A READ-MODIFY-WRITE SEQUENCE OF THE RLDA WORKS. BITS 15-0 ARE BIT CLEARED USING GROWING AND WALKING 0'S AND 1'S.

TEST 14 - BUS RESET OF RLCS

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL CLEAR ALL BITS OF THE RLCS WITH THE EXCEPTION OF BIT 7 (CONTROLLER READY), BIT 0 (DRIVE READY) AND BIT 15 (COMPOSITE ERROR) IF BIT 14 (DRIVE ERROR) IS SET.

TEST 15 - BUS RESET OF RLBA

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL CLEAR ALL BITS OF THE RLBA.

TEST 16 - BUS RESET OF RLDA

THIS TEST WILL VERIFY THAT THE BUS RESET OF THE PROCESSOR WILL CLEAR ALL BITS OF THE RLDA.

TEST 17 - UNIQUENESS OF RLCS

THIS TEST WILL VERIFY THAT WHEN THE RLCS (XXXXX0) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLBA AND THE RLDA ARE SET UP WITH KNOWN DATA, THE RLDA IS WRITTEN, THEN THE RLBA AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 18 - UNIQUENESS OF RLBA

THIS TEST WILL VERIFY THAT WHEN THE RLBA (XXXXX2) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLCS AND RLDA ARE WRITTEN WITH KNOWN DATA, THE RLBA IS WRITTEN, THEN THE RLCS AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 19 - UNIQUENESS OF RLDA

THIS TEST WILL VERIFY THAT WHEN THE RLDA (XXXXX4) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. BOTH THE RLCS AND RLBA ARE WRITTEN WITH KNOWN DATA, THE RLDA IS WRITTEN, THEN THE RLCS AND RLBA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 20 - UNIQUENESS OF RLMP

THIS TEST WILL VERIFY THAT WHEN THE RLMP (XXXXX6) IS ADDRESSED ONLY THAT REGISTER IS EFFECTED. THE RLCS, RLBA AND RLDA ARE WRITTEN WITH KNOWN DATA, THE RLMP IS WRITTEN, THEN THE RLCS, RLBA AND RLDA ARE VERIFIED THAT THEY DID NOT CHANGE.

TEST 21 - NOOP FUNCTION

THIS TEST WILL VERIFY THE OPERATION OF THE NOOP (0) FUNCTION ON PDP-11'S ONLY, SINCE ON AN LSI-11 IT IS A MAINTENANCE FUNCTION. THE ABILITY OF CONTROLLER READY TO RESET AND NO ERRORS ARE CHECKED.

TEST 22 - TEST NOOP DOES NOTHING

THIS TEST WILL CHECK THAT THE NOOP FUNCTION WILL NOT DISTURB ANY REGISTERS OF THE CONTROLLER.

TEST 23 - TEST OF INTERRUPT

THIS TEST WILL CAUSE AN INTERRUPT FROM THE CONTROLLER USING NOOP (RL11 ONLY) TO CHECK THE INTERRUPT LOGIC AND VECTOR.

TEST 24 - TEST PRIORITY BR LEVEL

THIS TEST WILL CHECK THAT THE PROPER PRIORITY IS ON THE BOARD. WE VERIFY THAT ABOVE THE LEVEL THE BOARD WILL NOT INTERRUPT AND BELOW IT, IT WILL.

TEST 25 - GET STATUS FUNCTION

THIS TEST WILL VERIFY THAT THE GET STATUS FUNCTION (2) WILL COMPLETE CORRECTLY. THE RLDA IS SET UP AND GET STATUS IS ISSUED. CONTROLLER READY IS CHECKED AS WELL AS ERROR BITS. (FIRST TEST A DRIVE MUST BE PRESENT.)

TEST 26 - GET STATUS FUNCTION INTERRUPT

THIS TEST WILL VERIFY THAT THE GET STATUS FUNCTION WILL GENERATE AN INTERRUPT ON COMPLETION.

TEST 27 - GET STATUS FUNCTION GENERATES OPI

THIS TEST WILL PROVE THE ABILITY FOR OPI (OPERATION INCOMPLETE) TO SET AND THAT THE DRIVE COMMAND IS BEING TRANSMITTED CORRECTLY. THE COMMAND WORD (RLDA) IS SET UP WITH THE MARKER BIT ONLY. AN OPI IS EXPECTED TO RESULT, THIS IS CHECKED.

TEST 28 - OPI UNDER INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF AN OPI TO CAUSE AN INTERRUPT TO OCCUR. WE SEND ONLY THE MARKER BIT WITH THE GET STATUS COMMAND AND EXPECT AN OPI ERROR.

TEST 30 - READ HEADER FUNCTION INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF THE READ HEADER FUNCTION TO INTERRUPT ON COMPLETION.

TEST 31 - REPEATED RD HDRS YIELD SAME CYL AND HD

THIS TEST WILL CHECK THAT ON REPEATED READ HEADERS THE CYLINDER AND HEAD BITS OF THE HEADER WORD (RLMP) ARE ALWAYS THE SAME.

TEST 32 - CHECK OF HEADER CRC

THIS TEST WILL VERIFY THE HEADER CRC THAT FOLLOWS THE TWO HEADER WORDS IS ACTUALLY THE CORRECT CRC-16 CALCULATION OF THE TWO HEADER WORDS.

TEST 33 - CHECK CONSECUTIVE HEADERS

THIS TEST WILL CHECK THAT HEADERS ARE CONSECUTIVE.

TEST 34 - SEEK FUNCTION

THIS TEST WILL CHECK THE SEEK FUNCTION (3) TO RESET CONTROLLER READY AND POST NO ERRORS. COMMAND WORD IS LOADED WITH A ONE CYLINDER FORWARD SEEK.

TEST 35 - CHECK DRIVE READY ON SEEK

THIS TEST WILL CHECK THAT DRIVE READY CLEARS AND RESETS ON ISSUANCE OF A SEEK COMMAND.

TEST 36 - SEEK FUNCTION INTERRUPT

THIS TEST WILL CHECK THE ABILITY OF A SEEK COMMAND TO GENERATE AN INTERRUPT ON CONTROLLER READY RESETTING AND NOT ONE ON DRIVE READY RESETTING.

TEST 37 - TEST DIFFERENCE WORD TRANSMISSION

THIS TEST WILL TRY TO VERIFY THAT BITS 14-7, 6, 2, 0 OF THE COMMAND WORD GET TRANSMITTED CORRECTLY. WE ISSUE SEEKS FROM TRACK 0 WITH COMMAND WORDS OF WALKING AND GROWING 0'S AND 1'S. ALL SEEKS ARE VERIFIED WITH A READ HEADER AND RETURN TO TRACK 0 BEFORE NEXT PATTERN IS ISSUED.

TEST 38 - VERIFY HEAD SELECT 0 VIA RD HEADER

THIS TEST WILL VERIFY THAT HEAD 0 CAN BE SELECTED AND READ VIA READ HEADER.

TEST 39 - VERIFY HEAD SELECT 1 VIA RD HEADER

THIS TEST WILL VERIFY THAT HEAD 1 CAN BE SELECTED AND READ VIA READ HEADER.

TEST 40 - VERIFY HEAD SELECT 0 VIA GET STATUS

THIS TEST WILL VERIFY THE WORD RETURNED TO THE RLMP BY A GET STATUS CONTAINS THE RIGHT HEAD SELECT.

TEST 41 - VERIFY HEAD SELECT 1 VIA GET STATUS

THIS TEST WILL VERIFY THE WORD RETURNED TO THE RLMP BY A GET STATUS CONTAINS THE RIGHT HEAD SELECT.

TEST 42 - TEST TIME AT WHICH DP WD GETS

THIS TEST WILL CHECK THAT THE DIFFERENCE WORD (RLDA) ACTUALLY DOES GET TRANSMITTED PRIOR TO CONTROLLER READY RESETTING. THIS IS DONE BY ISSUING A SEEK, WAITING FOR CONTROLLER READY AND RE-LOADING THE RLDA. THE SEEK IS THEN VERIFIED TO SEE IF IT IS CORRECT.

TEST 43 - EXTENSIVE CHECK OF CRC

THIS TEST WILL MORE EXTENSIVELY CHECK THE CRC LOGIC BY POSITIONING AT DIFFERENT POINTS ON THE PACK AND CHECKING THAT THE HEADER CRC RECEIVED IS CORRECT.

TEST 44 - VERIFY GET STATUS WHILE DRDY IS LOW

THIS TEST WILL CHECK THE ABILITY TO PERFORM A GET STATUS WHILE THE DRIVE IS SEEKING.

a

CZRLGCO RL11/RLV11 CTLR TST 1 MACRO V04.00 30-NOV-81 13:58:41
 TABLE OF CONTENTS

2-	76	GLOBAL DATA
2-	207	PATTERNS FOR DIFFERENCE WORD
3-	2	GLOBAL TEXT
4-	1	GLOBAL ERRORS
5-	2	LOAD PROTECTION TABLE
5-	9	INITIALIZATION CODE
5-	107	AUTO DROP SECTION
6-	2	GLOBAL SUBROUTINES
6-	24	ROUTINE TO CHECK FOR CONTROLLER ERRORS
6-	104	LOAD RLCS
6-	206	ROUTINE TO CALCULATE CRC
7-	1	**TEST 1** - RLCS ADDRESSABILITY
7-	26	**TEST 2** - RLBA ADDRESSABILITY
7-	52	**TEST 3** - RLDA ADDRESSABILITY
7-	77	**TEST 4** - RLMP ADDRESSABILITY
7-	102	**TEST 5** - READ WRITE OF RLCS
7-	144	**TEST 6** - READ WRITE OF RLBA
8-	11	**TEST 7** - READ WRITE OF RLDA
8-	44	**TEST 8** - BIS OF RLCS
8-	82	**TEST 9** - BIC OF RLCS
8-	118	**TEST 10** - BIS OF RLBA
8-	153	**TEST 11** - BIC OF RLBA
8-	185	**TEST 12** - BIS OF RLDA
8-	216	**TEST 13** - BIC OF RLDA
8-	248	**TEST 14** - BUS RESET OF RLCS
8-	284	**TEST 15** - BUS RESET OF RLBA
8-	310	**TEST 16** - BUS RESET OF RLDA
8-	333	**TEST 17** - UNIQUENESS OF RLCS
8-	375	**TEST 18** - UNIQUENESS OF RLBA
8-	417	**TEST 19** - UNIQUENESS OF RLDA
8-	461	**TEST 20** - UNIQUENESS OF RLMP
8-	514	**TEST 21** - NOOP FUNCTION(RL11 ONLY)
8-	543	**TEST 22** - TEST NOOP DOES NOTHING (RL11 ONLY)
8-	597	**TEST 23** - TEST OF INTERRUPT (RL11 ONLY)
8-	634	**TEST 24** - TEST PRIORITY BR LEVEL
8-	685	**TEST 25** - GET STATUS FUNCTION
8-	710	**TEST 26** - GET STATUS FUNCTION INTERRUPT
8-	743	**TEST 27** - GET STATUS FUNCTION GENERATES OPI W/O GS BIT
8-	773	**TEST 28** - OPI UNDER INTERRUPT
8-	807	**TEST 29** - READ HEADER FUNCTION
8-	823	**TEST 30** - READ HEADER FUNCTION INTERRUPT
8-	849	**TEST 31** - REPEATED RD HDRS YIELD SAME CYL AND HD
8-	897	**TEST 32** - CHECK OF HEADER CRC
8-	940	**TEST 33** - CHECK CONSECUTIVE HEADERS
8-	1014	**TEST 34** - SEEK FUNCTION
8-	1038	**TEST 35** - CHECK DRIVE READY ON SEEK
8-	1068	**TEST 36** - SEEK FUNCTION INTERRUPT
8-	1114	**TEST 37** - TEST DIFFERENCE WORD TRANSMISSION
8-	1237	**TEST 38** - VERIFY HEAD SELECT 0 VIA RD HDR
8-	1285	**TEST 39** - VERIFY HEAD SELECT 1 VIA RD HDR
8-	1332	**TEST 40** - VERIFY HEAD SELECT 0 VIA GET STATUS
8-	1379	**TEST 41** - VERIFY HEAD SELECT 1 VIA GET STATUS
8-	1427	**TEST 42** - TEST TIME AT WHICH DIF WD GETS TRANSMITTED
8-	1526	**TEST 43** - EXTENSIVE CHECK OF HEADER CRC
8-	1661	**TEST 44** - VERIFY GET STATUS WHILE DRDY IS LOW


```

1  .TITLE  SVC.MLB SOURCE FILE
2  .MACRO  MSMCHIGH
3  .MCALL  BCOMPL,BERROR,BGNAU,BGNAUTO,BGNCLN,BGNDU,BGNHRD,BGNHW,BGNINI,BGNMOD,BGNMSG,BGNPROT,BG
4  .MCALL  BGNSRV,BGNSUB,BGNSW,BGNTST,BNCOMP,BNERRO,BREAK,BRESET,CKLOOP,CLOCK,CLOSE,CLRVEC,COMME
5  .MCALL  DISPAT,DISPLA,DOCLN,DODU,DORPT,ENDAU,ENDAUTO,ENDCLN,ENDCOM,ENDDU,ENDHRD,ENDHW,ENDINI,
6  .MCALL  ENDRPT,ENDSEG,ENDSET,ENDSFT,ENDSRV,ENDSUB,ENDSW,ENDTST,EQUALS,ERRDF,ERRHRD,ERROR,ERRS
7  .MCALL  FEQUAL,GETBYTE,GETPRI,GETWORD,GMANIA,GMANID,GMANIL,GPHARD,GPRMA,GPRMD,GPRML,HEADER,IN
8  .MCALL  LASTAD,MANUAL,MEMORY,OPEN,POINTE,PRINTB,PRINTF,PRINTS,PRINTX,READBU,READEF,RFLAGS,SET
9  .MCALL  XFER,XFERF,XFERT
10 .MACRO  MSMCHIGH
11 .ENDM
12 .ENDM  MSMCHIGH
13 .MACRO  BCOMPLETE          LABEL
14 .MCALL  MSGNINS
15 MSGNINS <BCS LABEL>
16 .ENDM  BCOMPLETE
17 .MACRO  BERROR LABEL
18 .MCALL  MSGNINS
19 MSGNINS <BCS LABEL>
20 .ENDM  BERROR
21 .MACRO  BGNAU
22 .MCALL  MSPUSH,MSINCR,MSGNGBL
23 ISAU=FSBGN
24 MSPUSH TSNS,TSNESTLEV,FSAU
25 TSSAU=TSTAGNUM
26 MSINCR TSTAGNUM
27 MSGNGBL LSAU
28 .ENDM  BGNAU
29 .MACRO  BGNAUTO
30 .MCALL  MSPUSH,MSINCR,MSGNGBL
31 ISAUTO=FSBGN
32 MSPUSH TSNS,TSNESTLEV,FSAUTO
33 TSSAUTO=TSTAGNUM
34 MSINCR TSTAGNUM
35 MSGNGBL LSAUTO
36 .ENDM  BGNAUTO
37 .MACRO  BGNCLN
38 .MCALL  MSPUSH,MSINCR,MSGNGBL
39 ISCLN=FSBGN
40 MSPUSH TSNS,TSNESTLEV,FSCLEAN
41 TSSCLEAN=TSTAGNUM
42 MSINCR TSTAGNUM
43 MSGNGBL LSCLEAN
44 .ENDM  BGNCLN
45 .MACRO  BGNDU
46 .MCALL  MSPUSH,MSINCR,MSGNGBL
47 ISDU=FSBGN
48 MSPUSH TSNS,TSNESTLEV,FSDU
49 TSSDU=TSTAGNUM
50 MSINCR TSTAGNUM
51 MSGNGBL LSDU
52 .ENDM  BGNDU
53 .MACRO  BGNHRD
54 .MCALL  MSPUSH,MSINCR,MSGNINS,MSGNGBL
55 ISHRD=FSBGN
56 MSPUSH TSNS,TSNESTLEV,FSHARD
57 TSSHARD=TSTAGNUM

```

```
58 MSINCR T$TAGNUM
59 .IRP TAG,<\T$$HARD>
60 MSGNINS <.WORD L'TAG'-L$HARD/2>
61 .ENDM
62 MSGNGBL L$HARD
63 .ENDM BGNHRD
64 .MACRO BGNHW LABEL
65 .MCALL MSPUSH,MSINCR,MSGNINS,MSGNGBL
66 MSPUSH TSNS,T$NESTLEV,F$HW
67 T$$HW=T$TAGNUM
68 MSINCR T$TAGNUM
69 .IRP N,<\T$$HW>
70 MSGNINS <.WORD L'N-L$HW/2>
71 .ENDM
72 MSGNGBL L$HW
73 MSGNGBL LABEL
74 .ENDM BGNHW
75 .MACRO BGNINIT
76 .MCALL MSPUSH,MSINCR,MSGNGBL
77 I$INIT=F$BGN
78 MSPUSH TSNS,T$NESTLEV,F$INIT
79 T$$INIT=T$TAGNUM
80 MSINCR T$TAGNUM
81 MSGNGBL L$INIT
82 .ENDM BGNINIT
83 .MACRO BGNMOD ARG
84 .MCALL MSGNGBL,MSPUSH
85 .IF NE F$END-I$MOD
86 .ERROR ;'BGNMOD' IN MOD
87 .MEXIT
88 .IFF
89 I$MOD=F$BGN
90 .ENDC
91 .IF NB ARG
92 MSGNGBL ARG
93 .ENDC
94 MSPUSH TSNS,T$NESTLEV,F$MOD
95 .ENDM BGNMOD
96 .MACRO BGNMSG ARG
97 .MCALL MSPUSH,MSINCR,MSGNGBL
98 I$MSG=F$BGN
99 MSPUSH TSNS,T$NESTLEV,F$MSG
100 T$$MSG=T$TAGNUM
101 MSINCR T$TAGNUM
102 MSGNGBL ARG
103 .ENDM BGNMSG
104 .MACRO BGNPROT
105 .MCALL MSPUSH,MSINCR,MSGNGBL
106 I$PROT=F$BGN
107 MSPUSH TSNS,T$NESTLEV,F$PROT
108 T$$PROT=T$TAGNUM
109 MSINCR T$TAGNUM
110 MSGNGBL L$PROT
111 .ENDM BGNPROT
112 .MACRO BGNPTAB
113 .MCALL MSINCR,MSDECR,MSWORD,MSGNINS,MSGNTAG
114 .IF NE F$BGN-I$SETUP
```



```
115 .ERROR ; MISSING 'BGNSETUP'
116 .MEXIT
117 .ENDC
118 .IF NE FSEND-ISPTAB
119 .ERROR ; MISSING 'ENDPTAB'
120 .MEXIT
121 .ENDC
122 ISPTAB=FSBGN
123 TSPTAB=TSTAGNUM
124 MSINCR TSTAGNUM
125 MSDECR TSPCNT
126 .IF LE TSPCNT
127 MSWORD <0>
128 .IFF
129 .IRP N,<\TSPTAB>
130 MSGNINS <.WORD L'N'>
131 .ENDM
132 .ENDC
133 TSSDAT=TSTAGNUM
134 MSINCR TSTAGNUM
135 .IRP N,<\TSSDAT>
136 MSGNINS <.WORD L'N-./2-1'>
137 .ENDM
138 MSGNTAG L,TSSPTAB
139 TSSPTAB=TSPTAB
140 MSINCR TSPTNUM
141 .ENDM BGNPTAB
142 .MACRO BGNRPT
143 .MCALL MSPUSH,MSINCR,MSGNGBL
144 ISRPT=FSBGN
145 MSPUSH TSNS,TSNESTLEV,FSRPT
146 TSSRPT=TSTAGNUM
147 MSINCR TSTAGNUM
148 MSGNGBL LSRPT
149 .ENDM BGNRPT
150 .MACRO BGNSEG
151 .MCALL MSPUSH,MSINCR,MSSVC
152 ISSEG=FSBGN
153 MSPUSH TSNS,TSNESTLEV,FSSEG
154 TSSSEG=SSLSYM
155 MSPUSH TSSEK,TSSEGLEV,TSSSEG
156 MSINCR SSLSYM
157 MSSVC CSBSEG
158 .ENDM BGNSEG
159 .MACRO BGNSETUP ARG
160 .MCALL MSINCR
161 .IF NE FSEND - ISSETUP
162 .ERROR ; ONLY 1 SETUP!
163 .MEXIT
164 .ENDC
165 .IF NE TSLAST-1
166 .ERROR ; 'LASTAD' MUST PRECEDE 'BGNSETUP'
167 .MEXIT
168 .ENDC
169 ISSETUP=FSBGN
170 TSSPTAB=TSTAGNUM
171 MSINCR TSTAGNUM
```

```
172 .IF NB ARG
173 T$PCNT=ARG
174 T$$PC=ARG
175 .IFF
176 .ERROR ; MISSING # OF PTABLES
177 T$PCNT=1
178 .ENDC
179 .ENDM BGNSETUP
180 .MACRO BGNSFT
181 .MCALL MSPUSH,MSINCR,MSGNINS,MSGNGBL
182 ISSFT=F$BGN
183 MSPUSH TSNS,TSNESTLEV,F$SOFT
184 T$$SOFT=T$TAGNUM
185 MSINCR T$TAGNUM
186 .IRP TAG,<\T$$SOFT>
187 MSGNINS <.WORD L'TAG'-L$SOFT/2>
188 .ENDM
189 MSGNGBL L$SOFT
190 .ENDM BGNSFT
191 .MACRO BGNSRV ARG
192 .MCALL MSPUSH,MSINCR,MSGNGBL
193 ISSRV=F$BGN
194 MSPUSH TSNS,TSNESTLEV,F$SRV
195 T$$SRV=T$TAGNUM
196 MSINCR T$TAGNUM
197 MSGNGBL ARG
198 .ENDM BGNSRV
199 .MACRO BGNSUB
200 .MCALL MSINCR,MSGNSUB,MSPUSH,M$SVC
201 .IF NE F$END-I$SEG
202 .ERROR ;'BGNSUB' IN SEG
203 .MEXIT
204 .ENDC
205 .IF NE F$END-I$SUB
206 .ERROR ;'BGNSUB' IN SUB
207 .MEXIT
208 .ENDC
209 .IF NE F$BGN-I$TST
210 .ERROR ;'BGNSUB' NOT IN TST
211 .MEXIT
212 .ENDC
213 ISSUB=F$BGN
214 MSINCR T$SUBNUM
215 .RADIX 10
216 MSGNSUB \T$TESTNUM,\T$SUBNUM
217 .RADIX 8
218 MSPUSH TSNS,TSNESTLEV,F$SUB
219 T$$SUB=T$TAGNUM
220 MSINCR T$TAGNUM
221 M$SVC CS$SUB
222 .ENDM BGNSUB
223 .MACRO BGNSW LABEL
224 .MCALL MSPUSH,MSINCR,MSGNINS,MSGNGBL
225 MSPUSH TSNS,TSNESTLEV,F$SW
226 T$$SW=T$TAGNUM
227 MSINCR T$TAGNUM
228 .IRP N,<\T$$SW>
```



```

229 MSGNINS <.WORD L'N-LSSW/2>
230 .ENDM
231 MSGNGBL LSSW
232 MSGNGBL LABEL
233 .ENDM
234 .MACRO BGNTST NUM
235 .MCALL MSINCR,MSGNTST,MSPUSH,MSINCR
236 .IF NE FSEND-ISSUB
237 .ERROR ;'BGNTST' IN SUB
238 .MEXIT
239 .ENDC
240 .IF NE FSEND-ISSEG
241 .ERROR ;'BGNTST' IN SEG
242 .MEXIT
243 .ENDC
244 .IF NE FSEND-ISTST
245 .ERROR ;'BGNTST' IN TST
246 .MEXIT
247 .IFF
248 ISTST=F$BGN
249 .ENDC
250 .IF B NUM
251 MSINCR T$TESTNUM
252 .IFF
253 T$TESTNUM=NUM
254 .ENDC
255 T$SUBNUM=0
256 .RADIX 10
257 MSGNTST \T$TESTNUM
258 .RADIX 8
259 MSPUSH TSNS,TSNESTLEV,F$TEST
260 T$STEST=T$TAGNUM
261 MSINCR T$TAGNUM
262 T$TSTSEQ=1
263 .ENDM
264 .MACRO BNCOMPLETE LABEL
265 .MCALL MSGNINS
266 MSGNINS <BCC LABEL>
267 .ENDM
268 .MACRO BNERROR LABEL
269 .MCALL MSGNINS
270 MSGNINS <BCC LABEL>
271 .ENDM
272 .MACRO BREAK
273 .MCALL MSSVC
274 MSSVC C$BRK
275 .ENDM
276 .MACRO BRESET
277 .MCALL MSSVC
278 MSSVC C$RESET
279 .ENDM
280 .MACRO CKLOOP
281 .MCALL MSSVC
282 MSSVC C$CLP1
283 .ENDM
284 .MACRO CLOCK ARG1,ARG2
285 .MCALL MSLDRO,MSSVC,MSRNRO

```



```

343 .ENDM DEVTYP
344 .MACRO DISPATCH NUM
345 .MCALL MSWORD,MSGNGBL
346 .RADIX 10
347 MSWORD \NUM
348 MSGNGBL LSDISPATCH
349 T$TEMP=1
350 .REPT NUM
351 .IRP N,<\T$TEMP>
352 MSWORD T'N
353 .ENDM
354 T$TEMP=T$TEMP + 1
355 .ENDR
356 .RADIX 8
357 .ENDM DISPATCH
358 .MACRO DISPLAY ARG
359 .MCALL MSWORD
360 .IF NE ISHRD-F$BGN
361 .IF NE ISSFT-F$BGN
362 .ERROR ;'DISPLAY' MUST BE IN H.W. OR S.W. QUES.
363 .MEXIT
364 .ENDC
365 .ENDC
366 .IF B ARG
367 .ERROR ;MISSING 'DISPLAY' ARG.
368 .MEXIT
369 .ENDC
370 T$CODE=G$DISP
371 MSWORD <T$CODE>
372 MSWORD <ARG>
373 .ENDM DISPLAY
374 .MACRO DOCLN
375 .MCALL MSSVC
376 MSSVC C$DCLN
377 .ENDM DOCLN
378 .MACRO DODU ARG
379 .MCALL MSLDRO,MSSVC
380 MSLDRO ARG,DODU
381 MSSVC C$DODU
382 .ENDM DODU
383 .MACRO DORPT
384 .MCALL MSSVC
385 MSSVC C$DRPT
386 .ENDM DORPT
387 .MACRO ENDAU
388 .MCALL MSPOP,MSGNTAG,MSSVC,MSENDERR
389 MSPOP TSNS,TSNESTLEV,T$TEMP
390 .IF EQ F$AU-T$TEMP
391 MSGNTAG L,T$SAU
392 S$LSYM=T$LSYM
393 MSSVC C$AU
394 I$AU=F$SEND
395 .IFF
396 MSENDERR ENDAU,T$TEMP
397 .ENDC
398 .ENDM ENDAU
399 .MACRO ENDAUTO

```



```
457 S$LSYM=T$LSYM
458 I$HRD=F$END
459 .IFF
460 M$ENDERR          ENDHRD,T$TEMP
461 .ENDC
462 .ENDM  ENDHRD
463 .MACRO  ENDHW
464 .MCALL  M$POP,M$GNTAG,M$ENDERR
465 M$POP  T$NS,T$NESTLEV,T$TEMP
466 .IF EQ  F$HW-T$TEMP
467 M$GNTAG L,T$$HW
468 S$LSYM=T$LSYM
469 .IFF
470 M$ENDERR          ENDHW,T$TEMP
471 .ENDC
472 .ENDM  ENDHW
473 .MACRO  ENDINIT
474 .MCALL  M$POP,M$GNTAG,M$$VC,M$ENDERR
475 M$POP  T$NS,T$NESTLEV,T$TEMP
476 .IF EQ  F$INIT-T$TEMP
477 M$GNTAG L,T$$INIT
478 S$LSYM=T$LSYM
479 M$$VC  C$INIT
480 I$INIT=F$END
481 .IFF
482 M$ENDERR          ENDINIT,T$TEMP
483 .ENDC
484 .ENDM  ENDINIT
485 .MACRO  ENDMOD
486 .MCALL  M$POP,M$ENDERR
487 .IF EQ  F$BGN-I$MOD
488 I$MOD=F$END
489 .IFF
490 .ERROR  ; MISSING 'BGNMOD'
491 .MEXIT
492 .ENDC
493 M$POP  T$NS,T$NESTLEV,T$TEMP
494 .IF NE  F$MOD-T$TEMP
495 M$ENDERR          ENDMOD,T$TEMP
496 .ENDC
497 .ENDM  ENDMOD
498 .MACRO  ENDMSG
499 .MCALL  M$POP,M$GNTAG,M$$VC,M$ENDERR
500 M$POP  T$NS,T$NESTLEV,T$TEMP
501 .IF EQ  F$MSG-T$TEMP
502 M$GNTAG L,T$$MSG
503 S$LSYM=T$LSYM
504 M$$VC  C$MSG
505 I$MSG=F$END
506 .IFF
507 M$ENDERR          ENDMSG,T$TEMP
508 .ENDC
509 .ENDM  ENDMSG
510 .MACRO  ENDPROT
511 .MCALL  M$POP
512 M$POP  T$NS,T$NESTLEV,T$TEMP
513 .IF NE  F$PROT-T$TEMP
```

```
514 .ERROR ;MISSING BGNPROT
515 .MEXIT
516 .ENDC
517 .ENDM ENDPROT
518 .MACRO ENDPTAB
519 .MCALL MSGNTAG
520 .IF NE F$BGN - ISPTAB
521 .ERROR ; MISSING 'BGNPTAB'
522 .MEXIT
523 .ENDC
524 ISPTAB=F$END
525 MSGNTAG L,T$$DAT
526 .ENDM ENDPTAB
527 .MACRO ENDRPT
528 .MCALL M$POP,MSGNTAG,M$SVC,M$ENDERR
529 M$POP TSNS,TSNESTLEV,T$TEMP
530 .IF EQ F$RPT-T$TEMP
531 MSGNTAG L,T$$RPT
532 SLSYM=TSLSYM
533 M$SVC CSRPT
534 ISRPT=F$END
535 .IFF
536 M$ENDERR ENDRPT,T$TEMP
537 .ENDC
538 .ENDM ENDRPT
539 .MACRO ENDSEG
540 .MCALL M$POP,MSGNLS,M$SVC,M$ENDERR
541 M$POP TSNS,TSNESTLEV,T$TEMP
542 .IF EQ F$SEG-T$TEMP
543 M$POP TSSEK,TSSEGLEV,T$$SEG
544 MSGNLS T$$SEG
545 M$SVC C$SEG
546 .IF EQ T$SEGLEV+1
547 ISSEG=F$END
548 .ENDC
549 .IFF
550 M$ENDERR ENDSEG,T$TEMP
551 .ENDC
552 .ENDM ENDSEG
553 .MACRO ENDSETUP
554 .IF NE F$BGN - ISSETUP
555 .ERROR ; MISSING 'BGNSETUP'
556 .MEXIT
557 .ENDC
558 ISSETUP=F$END
559 .IF NE T$$PC - T$PTNUM
560 .ERROR ; PTABLE COUNT INCORRECT
561 .ENDC
562 T$FREE=.
563 T$SIZE=-.L$LAST / 2
564 T$PTHV=T$PTNUM
565 .ENDM ENDSETUP
566 .MACRO ENDSFT
567 .MCALL M$POP,MSGNINS,MSGNTAG,M$ENDERR
568 M$POP TSNS,TSNESTLEV,T$TEMP
569 .IF EQ F$SOFT-T$TEMP
570 MSGNINS .EVEN
```



```
571 MSGNTAG L,T$$$SOFT
572 SSLSYM=T$LSYM
573 ISSFT=F$END
574 .IFF
575 M$ENDERR          ENDSFT,T$TEMP
576 .ENDC
577 .ENDM  ENDSFT
578 .MACRO ENDSRV ARG
579 .MCALL M$POP,MSGNTAG,MSGNINS,M$ENDERR
580 M$POP  TSNS,TSNESTLEV,T$TEMP
581 .IF EQ F$SRV-T$TEMP
582 MSGNTAG L,T$$$SRV
583 SSLSYM=T$LSYM
584 .IF NB ARG
585 MSGNINS <BICB #340,2(SP)>
586 MSGNINS <BISB ARG,2(SP)>
587 .ENDC
588 MSGNINS RTI
589 ISSRV=F$END
590 .IFF
591 M$ENDERR          ENDSRV,T$TEMP
592 .ENDC
593 .ENDM  ENDSRV
594 .MACRO ENDSUB
595 .MCALL M$POP,MSGNTAG,M$SVC,M$ENDERR
596 .IF EQ F$BGN-I$SUB
597 I$SUB=F$END
598 .IFF
599 .ERROR ;MISSING 'BGNSUB'
600 .MEXIT
601 .ENDC
602 M$POP  TSNS,TSNESTLEV,T$TEMP
603 .IF EQ F$SUB-T$TEMP
604 MSGNTAG L,T$$$SUB
605 SSLSYM=T$LSYM
606 M$SVC  C$ESUB
607 I$SUB=F$END
608 .IFF
609 M$ENDERR          ENDSUB,T$TEMP
610 .ENDC
611 .ENDM  ENDSUB
612 .MACRO ENDSW
613 .MCALL M$POP,MSGNTAG,M$ENDERR
614 M$POP  TSNS,TSNESTLEV,T$TEMP
615 .IF EQ F$SW-T$TEMP
616 MSGNTAG L,T$$$SW
617 SSLSYM=T$LSYM
618 .IFF
619 M$ENDERR          ENDSW,T$TEMP
620 .ENDC
621 .ENDM  ENDSW
622 .MACRO ENDTST
623 .MCALL M$POP,MSGNTAG,M$SVC,M$ENDERR
624 .IF EQ F$BGN-I$TST
625 I$TST=F$END
626 .IFF
627 .ERROR ;MISSING 'BGNTST'
```

```

628 .MEXIT
629 .ENDC
630 M$POP TSNS,TSNESTLEV,T$TEMP
631 .IF EQ F$TEST-T$TEMP
632 MSGNTAG L,T$$TEST
633 S$LSYM=T$LSYM
634 M$SVC C$ETST
635 I$TST=F$SEND
636 .IFF
637 M$SENDERR ENDTST,T$TEMP
638 .ENDC
639 .ENDM ENDTST
640 .MACRO EQUALS
641 .LIST
642 :
643 : BIT DIFINITIONS
644 :
645 BIT15== 10000
646 BIT14== 4000
647 BIT13== 2000
648 BIT12== 1000
649 BIT11== 400
650 BIT10== 200
651 BIT09== 100
652 BIT08== 40
653 BIT07== 20
654 BIT06== 10
655 BIT05== 4
656 BIT04== 2
657 BIT03== 1
658 BIT02== 4
659 BIT01== 2
660 BIT00== 1
661 :
662 BIT9== BIT09
663 BIT8== BIT08
664 BIT7== BIT07
665 BIT6== BIT06
666 BIT5== BIT05
667 BIT4== BIT04
668 BIT3== BIT03
669 BIT2== BIT02
670 BIT1== BIT01
671 BIT0== BIT00
672 :
673 : EVENT FLAG DEFINITIONS
674 : EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
675 :
676 EF.START== 32. : START COMMAND WAS ISSUED
677 EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
678 EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
679 EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
680 EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED
681 :
682 :
683 : PRIORITY LEVEL DEFINITIONS
684 :

```



```

685      PRI07== 340
686      PRI06== 300
687      PRI05== 240
688      PRI04== 200
689      PRI03== 140
690      PRI02== 100
691      PRI01== 40
692      PRI00== 0
693      ;
694      ;OPERATOR FLAG BITS
695      ;
696      EVL==      4
697      LOT==     10
698      ADR==     20
699      IDU==     40
700      ISR==    100
701      UAM==    200
702      BOE==    400
703      PNT==   1000
704      PRI==   2000
705      IXE==   4000
706      IBE==  10000
707      IER==  20000
708      LOE==  40000
709      HOE== 100000
710      .NLIST
711      .ENDM      EQUALS
712      .MACRO    ERRDF      NUM,MSGADR,PNTR
713      .MCALL    MSERRINS
714      MSERRINS      NUM,MSGADR,PNTR,CSERDF
715      .ENDM      ERRDF
716      .MACRO    ERRHRD     NUM,MSGADR,PNTR
717      .MCALL    MSERRINS
718      MSERRINS      NUM,MSGADR,PNTR,CSERHRD
719      .ENDM      ERRHRD
720      .MACRO    ERROR
721      .MCALL    MSSVC
722      MSSVC      CSERROR
723      .ENDM      ERROR
724      .MACRO    ERRSF      NUM,MSGADR,PNTR
725      .MCALL    MSERRINS
726      MSERRINS      NUM,MSGADR,PNTR,CSERSF
727      .ENDM      ERRSF
728      .MACRO    ERRSOFT   NUM,MSGADR,PNTR
729      .MCALL    MSERRINS
730      MSERRINS      NUM,MSGADR,PNTR,CSERSOFT
731      .ENDM      ERRSOFT
732      .MACRO    ERRTBL
733      .MCALL    MSGNGBL
734      MSGNGBL   L$ERRTBL
735      .LIST
736      ERRTP::      .WORD    0
737      ERRNBR::     .WORD    0
738      ERRMSG::     .WORD    0
739      ERRBLK::     .WORD    0
740      .NLIST
741      .ENDM      ERRTBL

```

```

742 .MACRO ESCAPE NAME
743 .MCALL MSSVC,MSGGETTOP,MSESCAPE,MSESCSEG
744 TSFLAG=FSEND
745 .IF IDN <TST>,<NAME>
746 T$TEMP=T$$TEST
747 TSFLAG=I$TST
748 .ENDC
749 .IF IDN <SUB>,<NAME>
750 T$TEMP=T$$SUB
751 TSFLAG=I$SUB
752 .ENDC
753 .IF IDN <SEG>,<NAME>
754 T$TEMP=T$$SEG
755 TSFLAG=I$SEG
756 .ENDC
757 .IF EQ TSFLAG-F$BGN
758 MSSVC C$ESCAPE
759 .IF IDN <SEG>,<NAME>
760 MSGGETTOP TSSEK,TSSEGLEV,T$TEMP
761 MSESCSEG \T$TEMP
762 .IFF
763 MSESCAPE \T$TEMP
764 .ENDC
765 .MEXIT
766 .ENDC
767 .ERROR :ILL. 'ESCAPE NAME' STMT.
768 .ENDM ESCAPE
769 .MACRO EXIT NAME
770 .MCALL MSSVC,MSEXIT,MSEXSEG,MSEXTJ,MSGGETTOP,MSCHECK,MSWORD,XFER
771 TSFLAG=FSEND
772 MSCHECK NAME,T$TEMP,TSFLAG
773 .IF EQ TSFLAG-F$JMP
774 MSWORD JSJMP
775 MSEXTJ \T$TEMP
776 .MEXIT
777 .ENDC
778 .IF EQ TSFLAG-F$BGN
779 MSSVC C$EXIT
780 .IF IDN <SEG>,<NAME>
781 MSGGETTOP TSSEK,TSSEGLEV,T$TEMP
782 MSEXSEG \T$TEMP
783 .IFF
784 MSEXIT \T$TEMP
785 .ENDC
786 .MEXIT
787 .ENDC
788 .IF IDN <HRD>,<NAME>
789 MSGGETTOP TSNS,TSNESTLEV,T$TEMP
790 .IF EQ F$HARD-T$TEMP
791 .IRP N,<\T$$HARD>
792 XFER L'N
793 .ENDM
794 .MEXIT
795 .ENDC
796 .ENDC
797 .IF IDN <SFT>,<NAME>
798 MSGGETTOP TSNS,TSNESTLEV,T$TEMP

```



```

799      .IF EQ FSSOFT-T$TEMP
800      .IRP  N,<\T$$$SOFT>
801      XFER  L'N
802      .ENDM
803      .MEXIT
804      .ENDC
805      .ENDC
806      .ERROR :ILL. "EXIT NAME" STMT.
807      .ENDM EXIT
808      .MACRO FEQUAL
809      .LIST
810      :
811      : FUNCTION-LEVEL I/O DEFINITIONS
812      :
813      Q.IOFN==      2
814      Q.IOLN==      4
815      Q.IOEF==      6
816      Q.IOSB==     10
817      Q.IOAE==     12
818      Q.IOPL==     14
819      IS.SUC==      1
820      .NLIST
821      .ENDM FEQUAL
822      .MACRO GETBYTE ARG
823      .MCALL MSSVC,MSRBRO
824      MSSVC CSGETB
825      MSRBRO ARG,GETBYTE
826      .ENDM GETBYTE
827      .MACRO GETWORD ARG
828      .MCALL MSSVC,MSRNRO
829      MSSVC CSGETW
830      MSRNRO ARG,GETWORD
831      .ENDM GETWORD
832      .MACRO GETPRI ARG
833      .MCALL MSSVC,MSRNRO
834      MSSVC CSGPRI
835      MSRNRO ARG,GETPRI
836      .ENDM GETPRI
837      .MACRO GMANIA $MSGADR,$DATADR,$RADIX,$LOLIM,$HILIM,$DFLT
838      .MCALL MSSVC,MSGENBR,MSWORD,MSGNLS,MSINCR,GPRMA
839      MSSVC CSGMAN
840      MSGENBR BR,\SSLSYM,$
841      MSWORD $DATADR
842      GPRMA $MSGADR,0,$RADIX,<$LOLIM>,<$HILIM>,$DFLT
843      MSGNLS
844      MSINCR SSLSYM
845      .ENDM GMANIA
846      .MACRO GMANID $MSGADR,$DATADR,$RADIX,$MASK,$LOLIM,$HILIM,$DFLT
847      .MCALL MSSVC,MSGENBR,MSWORD,MSGNLS,MSINCR,GPRMD
848      TSGMANID=1
849      MSSVC CSGMAN
850      MSGENBR BR,\SSLSYM,$
851      MSWORD $DATADR
852      GPRMD $MSGADR,0,$RADIX,$MASK,<$LOLIM>,<$HILIM>,$DFLT
853      MSGNLS
854      MSINCR SSLSYM
855      TSGMANID=0

```

```

: DIRECTIVE FUNCTION
: DIRECTIVE LOG UNIT NO.
: DIRECTIVE EVENT FLAG
: DIRECTIVE STATUS BLOCK
: DIRECTIVE AST ENTRY
: DIRECTIVE PARAMETER LIST
: REQUEST WAS SUCCESSFUL

```

```

856 .ENDM GMANID
857 .MACRO GMANIL $MSGADR,$DATADR,$MASK,$DFLT
858 .MCALL MSSVC,$MSGENBR,$MSWORD,$MSGNLS,$MSINCR,GPRML
859 MSSVC CS$GMAN
860 $MSGENBR BR,\SSLSYM,$
861 $MSWORD $DATADR
862 GPRML $MSGADR,0,$MASK,$DFLT
863 $MSGNLS
864 $MSINCR SSLSYM
865 .ENDM GMANIL
866 .MACRO GPHARD ARG1,ARG2
867 .MCALL MSLDRO,MSSVC,$MSRNRO
868 MSLDRO ARG1,GPHARD
869 MSSVC CS$GPHRD
870 $MSRNRO ARG2,GPHARD
871 .ENDM GPHARD
872 .MACRO GPRMA $MSGADR,$OFFSET,$RADIX,$LOLIM,$HILIM,$DFLT,$COUNT
873 .MCALL $MSRADIX,$MSDEFAULT,$MSEXCP,$MSWORD,$MSCNTOP
874 .IF IDN A,$RADIX
875 .ERROR ;INVALID RADIX
876 .ENDC
877 T$TEMP=$OFFSET&1
878 .IF NE T$TEMP & 1
879 .ERROR ;ODD OFFSET
880 .ENDC
881 .IF LT G$OFFSIZE-$OFFSET
882 .ERROR ;OFFSET TOO BIG
883 .ENDC
884 T$CODE=G$PRMA + <$OFFSET * G$OFFSET>
885 $MSRADIX $RADIX,T$TEMP
886 T$CODE=T$CODE ! T$TEMP
887 $MSDEFAULT $DFLT,T$TEMP
888 T$CODE=T$CODE ! T$TEMP
889 T$EXCP=0
890 $MSEXCP T$CODE,T$EXCP,G$LOLIM,T$LOLIM,$LOLIM
891 $MSEXCP T$CODE,T$EXCP,G$HILIM,T$HILIM,$HILIM
892 $MSCNTOP <$COUNT>,<T$CODE>
893 $MSWORD <T$CODE,$MSGADR,T$LOLIM,T$HILIM>
894 .IF NE T$EXCP
895 $MSWORD T$EXCP
896 .ENDC
897 .IF NB $COUNT
898 $MSWORD <$COUNT/2>
899 .ENDC
900 .ENDM GPRMA
901 .MACRO GPRMD $MSGADR,$OFFSET,$RADIX,$MASK,$LOLIM,$HILIM,$DFLT,$COUNT
902 .MCALL $MSRADIX,$MSDEFAULT,$MSEXCP,$MSWORD,$MSCNTOP
903 .IF IDN A,$RADIX
904 .IF EQ T$GMANID
905 .ERROR ;ASCII RADIX VALID ONLY ON 'GMANID'
906 .ENDC
907 .ENDC
908 T$TEMP=$OFFSET&1
909 .IF NE T$TEMP & 1
910 .ERROR ;ODD OFFSET
911 .ENDC
912 .IF LT G$OFFSIZE-$OFFSET

```



```

913 .ERROR ;OFFSET TOO BIG
914 .ENDC
915 TSCODE=GSPRMD + <$OFFSET * G$OFFSET>
916 MSRADIX $RADIX,T$TEMP
917 TSCODE=TSCODE ! T$TEMP
918 MSDEFAULT $DFLT,T$TEMP
919 TSCODE=TSCODE ! T$TEMP
920 T$EXCP=0
921 M$EXCP TSCODE,T$EXCP,G$LOLIM,T$LOLIM,$LOLIM
922 M$EXCP TSCODE,T$EXCP,G$HILIM,T$HILIM,$HILIM
923 MSCNTOP <$COUNT>,<T$CODE>
924 MSWORD <T$CODE,$MSGADR,$MASK,T$LOLIM,T$HILIM>
925 .IF NE T$EXCP
926 MSWORD T$EXCP
927 .ENDC
928 .IF NB $COUNT
929 MSWORD <$COUNT/2>
930 .ENDC
931 .ENDM GPRMD
932 .MACRO GPRML $MSGADR,$OFFSET,$MASK,$DFLT,$COUNT
933 .MCALL MSRADIX,MSDEFAULT,MSWORD,MSCNTOP
934 T$TEMP=$OFFSET&1
935 .IF NE T$TEMP & 1
936 .ERROR ;ODD OFFSET
937 .ENDC
938 .IF LT G$OFSIZE-$OFFSET
939 .ERROR ;OFFSET TOO BIG
940 .ENDC
941 TSCODE=GSPRML + <$OFFSET * G$OFFSET>
942 MSRADIX L,T$TEMP
943 TSCODE=TSCODE ! T$TEMP
944 MSDEFAULT $DFLT,T$TEMP
945 TSCODE=TSCODE ! T$TEMP
946 MSCNTOP <$COUNT>,<T$CODE>
947 MSWORD <T$CODE,$MSGADR,$MASK>
948 .IF NB $COUNT
949 MSWORD <$COUNT/2>
950 .ENDC
951 .ENDM GPRML
952 .MACRO HEADER FILNAM,REVLEV,DEPO,LOGST,DIAGTYPE,DIAGPRIO
953 .MCALL MSHAPT,MSHAP
954 MSHAPT FILNAM,REVLEV,DEPO,LOGST,DIAGTYPE,DIAGPRIO
955 MSHAP
956 .IF EQ 0$POINTER
957 .ERROR ;'POINTER' MUST PRECEDE 'HEADER'
958 .ENDC
959 .ENDM HEADER
960 .MACRO INLOOP
961 .MCALL M$SVC
962 M$SVC C$INLP
963 .ENDM INLOOP
964 .MACRO IOSETUP FUNC,LOGICAL,EVENT,IOSTAT,IOAST,PARAMS
965 .MCALL M$BYTE,MSWORD,MSIOSET
966 M$BYTE <1,12.>
967 MSWORD <FUNC,LOGICAL>
968 M$BYTE <0,0>
969 MSWORD <IOSTAT,IOAST>

```

```
970 MSIOSET PARAMS
971 .ENDM IOSETUP
972 .MACRO IOSTART ADR,ERR
973 .MCALL MSGNINS,MSSVC,MSSTART
974 MSGNINS <MOV ADR,-(SP)>
975 MSSVC CSQIO
976 .IF NB ERR
977 MSSTART ERR
978 .ENDC
979 .ENDM IOSTART
980 .MACRO KT11 DPAGE,SUP,USE
981 .LIST
982 .SBTTL MEMORY MANAGEMENT DEFINITIONS
983
984 ;*KT11 VECTOR ADDRESS
985
986 MMVEC= 250
987
988 ;*KT11 STATUS REGISTER ADDRESSES
989
990 SR0= 177572
991 SR1= 177574
992 SR2= 177576
993 SR3= 172516
994 .IF NB USE
995
996 ;*USER 'I' PAGE DESCRIPTOR REGISTERS
997
998 UIPDR0= 177600
999 UIPDR1= 177602
1000 UIPDR2= 177604
1001 UIPDR3= 177606
1002 UIPDR4= 177610
1003 UIPDR5= 177612
1004 UIPDR6= 177614
1005 UIPDR7= 177616
1006 .IF NB DPAGE
1007
1008 ;*USER 'D' PAGE DESCRIPTOR REGISTORS
1009
1010 UDPDR0= 177620
1011 UDPDR1= 177622
1012 UDPDR2= 177624
1013 UDPDR3= 177626
1014 UDPDR4= 177630
1015 UDPDR5= 177632
1016 UDPDR6= 177634
1017 UDPDR7= 177636
1018 .ENDC
1019
1020 ;*USER 'I' PAGE ADDRESS REGISTERS
1021
1022 UIPAR0= 177640
1023 UIPAR1= 177642
1024 UIPAR2= 177644
1025 UIPAR3= 177646
1026 UIPAR4= 177650
```



```
1027      UIPAR5= 177652
1028      UIPAR6= 177654
1029      UIPAR7= 177656
1030      .IF NB DPAGE
1031
1032      ;*USER 'D' PAGE ADDRESS REGISTERS
1033
1034      UDPAR0= 177660
1035      UDPAR1= 177662
1036      UDPAR2= 177664
1037      UDPAR3= 177666
1038      UDPAR4= 177670
1039      UDPAR5= 177672
1040      UDPAR6= 177674
1041      UDPAR7= 177676
1042      .ENDC
1043      .ENDC
1044      .IF NB SUP
1045
1046      ;*SUPERVISOR 'I' PAGE DESCRIPTOR REGISTERS
1047
1048      SIPDR0= 172200
1049      SIPDR1= 172202
1050      SIPDR2= 172204
1051      SIPDR3= 172206
1052      SIPDR4= 172210
1053      SIPDR5= 172212
1054      SIPDR6= 172214
1055      SIPDR7= 172216
1056      .IF NB DPAGE
1057
1058      ;*SUPERVISOR 'D' PAGE DESCRIPTOR REGISTERS
1059
1060      SDPDR0= 172220
1061      SDPDR1= 172222
1062      SDPDR2= 172224
1063      SDPDR3= 172226
1064      SDPDR4= 172230
1065      SDPDR5= 172232
1066      SDPDR6= 172234
1067      SDPDR7= 172236
1068      .ENDC
1069
1070      ;*SUPERVISOR 'I' PAGE ADDRESS REGISTERS
1071
1072      SIPAR0= 172240
1073      SIPAR1= 172242
1074      SIPAR2= 172244
1075      SIPAR3= 172246
1076      SIPAR4= 172250
1077      SIPAR5= 172252
1078      SIPAR6= 172254
1079      SIPAR7= 172256
1080      .IF NB DPAGE
1081
1082      ;*SUPERVISOR 'D' PAGE ADDRESS REGISTERS
1083
```

```
1084 SDPAR0= 172260
1085 SDPAR1= 172262
1086 SDPAR2= 172264
1087 SDPAR3= 172266
1088 SDPAR4= 172270
1089 SDPAR5= 172272
1090 SDPAR6= 172274
1091 SDPAR7= 172276
1092 .ENDC
1093 .ENDC
1094
1095 ;*KERNEL 'I' PAGE DESCRIPTOR REGISTERS
1096
1097 KIPDR0= 172300
1098 KIPDR1= 172302
1099 KIPDR2= 172304
1100 KIPDR3= 172306
1101 KIPDR4= 172310
1102 KIPDR5= 172312
1103 KIPDR6= 172314
1104 KIPDR7= 172316
1105 .IF NB DPAGE
1106
1107 ;*KERNEL 'D' PAGE DESCRIPTOR REGISTERS
1108
1109 KDPDR0= 172320
1110 KDPDR1= 172322
1111 KDPDR2= 172324
1112 KDPDR3= 172326
1113 KDPDR4= 172330
1114 KDPDR5= 172332
1115 KDPDR6= 172334
1116 KDPDR7= 172336
1117 .ENDC
1118
1119 ;*KERNEL 'I' PAGE ADDRESS REGISTERS
1120
1121 KIPAR0= 172340
1122 KIPAR1= 172342
1123 KIPAR2= 172344
1124 KIPAR3= 172346
1125 KIPAR4= 172350
1126 KIPAR5= 172352
1127 KIPAR6= 172354
1128 KIPAR7= 172356
1129 .IF NB DPAGE
1130
1131 ;*KERNEL 'D' PAGE ADDRESS REGISTERS
1132
1133 KDPAR0= 172360
1134 KDPAR1= 172362
1135 KDPAR2= 172364
1136 KDPAR3= 172366
1137 KDPAR4= 172370
1138 KDPAR5= 172372
1139 KDPAR6= 172374
1140 KDPAR7= 172376
```



```
1141 .ENDC
1142
1143 .NLIST
1144 .ENDM KT11
1145 .MACRO LASTAD
1146 .MCALL MSGNINS,MSGNGBL
1147 MSGNINS .EVEN
1148 TSLAST=1
1149 .IF EQ OSSETUP
1150 MSWORD <0>
1151 MSWORD <0>
1152 .IFF
1153 MSGNINS <.WORD T$FREE>
1154 MSGNINS <.WORD T$SIZE>
1155 .ENDC
1156 SVCGBL=0
1157 MSGNGBL L$LAST
1158 TSLTNO=T$TESTNUM
1159 .ENDM LASTAD
1160 .MACRO MANUAL
1161 .MCALL MSSVC
1162 MSSVC C$MANI
1163 .ENDM MANUAL
1164 .MACRO MEMORY ARG
1165 .MCALL MSSVC,MSRNRO
1166 MSSVC C$MEM
1167 MSRNRO ARG,MEMORY
1168 .ENDM MEMORY
1169 .MACRO OPEN ARG
1170 .MCALL MSLDRO,MSSVC
1171 MSLDRO ARG,OPEN
1172 MSSVC C$OPEN
1173 .ENDM OPEN
1174 .MACRO POINTER P1,P2,P3,P4,P5,P6,P7,P8,P9
1175 .IRP PX,<P1,P2,P3,P4,P5,P6,P7,P8,P9>
1176 .IF IDN <ALL>,<PX>
1177 OSBGNSFT=1
1178 OSBGNRPT=1
1179 OSGNSW=1
1180 OSAPTS=1
1181 OSAU=1
1182 OSDU=1
1183 OSERRTBL=1
1184 OSSETUP=1
1185 OSPOINTER=1
1186 .MEXIT
1187 .ENDC
1188 .IF IDN <NONE>,<PX>
1189 OSBGNSFT=0
1190 OSBGNRPT=0
1191 OSGNSW=0
1192 OSAPTS=0
1193 OSAU=0
1194 OSDU=0
1195 OSERRTBL=0
1196 OSSETUP=0
1197 OSPOINTER=1
```

```
1198 .MEXIT
1199 .ENDC
1200 .IF IDN <BGNSFT>,<PX>
1201 OSBGNSFT=1
1202 OSPOINTER=1
1203 .ENDC
1204 .IF IDN <BGNRPT>,<PX>
1205 OSBGNRPT=1
1206 OSPOINTER=1
1207 .ENDC
1208 .IF IDN <BGNSW>,<PX>
1209 OSGNSW=1
1210 OSPOINTER=1
1211 .ENDC
1212 .IF IDN <APTSTAT>,<PX>
1213 OSAPTS=1
1214 OSPOINTER=1
1215 .ENDC
1216 .IF IDN <BGNAU>,<PX>
1217 OSAU=1
1218 OSPOINTER=1
1219 .ENDC
1220 .IF IDN <BGNDU>,<PX>
1221 OSDU=1
1222 OSPOINTER=1
1223 .ENDC
1224 .IF IDN <ERRTBL>,<PX>
1225 OSERRTBL=1
1226 OSPOINTER=1
1227 .ENDC
1228 .IF IDN <BGNSETUP>,<PX>
1229 OSSETUP=1
1230 OSPOINTER=1
1231 .ENDC
1232 .ENDM
1233 .IF EQ OSPOINTER
1234 .ERROR ;ILL. ARGS. ON 'POINTER'
1235 .ENDC
1236 .ENDM POINTER
1237 .MACRO PRINTB FORM,P1,P2,P3,P4,P5,P6,P7,P8
1238 .MCALL MSPRINT
1239 .IF B FORM
1240 .ERROR ;MISSING FORMAT
1241 .MEXIT
1242 .ENDC
1243 .IF B <P1>
1244 MSPRINT C$PNTB,FORM
1245 .MEXIT
1246 .ENDC
1247 .IF B <P2>
1248 MSPRINT C$PNTB,FORM,<P1>
1249 .MEXIT
1250 .ENDC
1251 .IF B <P3>
1252 MSPRINT C$PNTB,FORM,<P1>,<P2>
1253 .MEXIT
1254 .ENDC
```



```
1255 .IF B <P4>
1256 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>
1257 .MEXIT
1258 .ENDC
1259 .IF B <P5>
1260 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>,<P4>
1261 .MEXIT
1262 .ENDC
1263 .IF B <P6>
1264 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>,<P4>,<P5>
1265 .MEXIT
1266 .ENDC
1267 .IF B <P7>
1268 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>
1269 .MEXIT
1270 .ENDC
1271 .IF B <P8>
1272 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>
1273 .MEXIT
1274 .ENDC
1275 M$PRINT C$PNTB,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1276 .ENDM PRINTB
1277 .MACRO PRINTF FORM,P1,P2,P3,P4,P5,P6,P7,P8
1278 .M$CALL M$PRINT
1279 .IF B FORM
1280 .ERROR :MISSING FORMAT
1281 .MEXIT
1282 .ENDC
1283 .IF B <P1>
1284 M$PRINT C$PNTF,FORM
1285 .MEXIT
1286 .ENDC
1287 .IF B <P2>
1288 M$PRINT C$PNTF,FORM,<P1>
1289 .MEXIT
1290 .ENDC
1291 .IF B <P3>
1292 M$PRINT C$PNTF,FORM,<P1>,<P2>
1293 .MEXIT
1294 .ENDC
1295 .IF B <P4>
1296 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>
1297 .MEXIT
1298 .ENDC
1299 .IF B <P5>
1300 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>,<P4>
1301 .MEXIT
1302 .ENDC
1303 .IF B <P6>
1304 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>,<P4>,<P5>
1305 .MEXIT
1306 .ENDC
1307 .IF B <P7>
1308 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>
1309 .MEXIT
1310 .ENDC
1311 .IF B <P8>
```

```
1312 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>
1313 .MEXIT
1314 .ENDC
1315 M$PRINT C$PNTF,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1316 .ENDM PRINTF
1317 .MACRO PRINTS FORM,P1,P2,P3,P4,P5,P6,P7,P8
1318 .MCALL M$PRINT
1319 .IF B FORM
1320 .ERROR ;MISSING FORMAT
1321 .MEXIT
1322 .ENDC
1323 .IF B <P1>
1324 M$PRINT C$PNTS,FORM
1325 .MEXIT
1326 .ENDC
1327 .IF B <P2>
1328 M$PRINT C$PNTS,FORM,<P1>
1329 .MEXIT
1330 .ENDC
1331 .IF B <P3>
1332 M$PRINT C$PNTS,FORM,<P1>,<P2>
1333 .MEXIT
1334 .ENDC
1335 .IF B <P4>
1336 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>
1337 .MEXIT
1338 .ENDC
1339 .IF B <P5>
1340 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>,<P4>
1341 .MEXIT
1342 .ENDC
1343 .IF B <P6>
1344 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>,<P4>,<P5>
1345 .MEXIT
1346 .ENDC
1347 .IF B <P7>
1348 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>
1349 .MEXIT
1350 .ENDC
1351 .IF B <P8>
1352 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>
1353 .MEXIT
1354 .ENDC
1355 M$PRINT C$PNTS,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1356 .ENDM PRINTS
1357 .MACRO PRINTX FORM,P1,P2,P3,P4,P5,P6,P7,P8
1358 .MCALL M$PRINT
1359 .IF B FORM
1360 .ERROR ;MISSING FORMAT
1361 .MEXIT
1362 .ENDC
1363 .IF B <P1>
1364 M$PRINT C$PNTX,FORM
1365 .MEXIT
1366 .ENDC
1367 .IF B <P2>
1368 M$PRINT C$PNTX,FORM,<P1>
```



```

1369 .MEXIT
1370 .ENDC
1371 .IF B <P3>
1372 M$PRINT C$PNTX,FORM,<P1>,<P2>
1373 .MEXIT
1374 .ENDC
1375 .IF B <P4>
1376 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>
1377 .MEXIT
1378 .ENDC
1379 .IF B <P5>
1380 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>,<P4>
1381 .MEXIT
1382 .ENDC
1383 .IF B <P6>
1384 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>,<P4>,<P5>
1385 .MEXIT
1386 .ENDC
1387 .IF B <P7>
1388 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>
1389 .MEXIT
1390 .ENDC
1391 .IF B <P8>
1392 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>
1393 .MEXIT
1394 .ENDC
1395 M$PRINT C$PNTX,FORM,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1396 .ENDM PRINTX
1397 .MACRO READBUS
1398 .M$CALL MSSVC
1399 MSSVC C$RDBU
1400 .ENDM READBUS
1401 .MACRO READEF EFN
1402 .M$CALL M$LDRO,MSSVC
1403 M$LDRO EFN,READEF
1404 MSSVC C$RÉFG
1405 .ENDM READEF
1406 .MACRO RFLAGS ARG
1407 .M$CALL MSSVC,M$RNRO
1408 MSSVC C$RFLA
1409 M$RNRO ARG,RFLAGS
1410 .ENDM RFLAGS
1411 .MACRO SETPRI ARG
1412 .M$CALL M$LDRO,MSSVC
1413 M$LDRO ARG,SETPRI
1414 MSSVC C$SPRI
1415 .ENDM SETPRI
1416 .MACRO SETVEC ARG1,ARG2,ARG3
1417 .M$CALL M$PUT,MSSVC,M$GNINS
1418 M$PUT #3,ARG1,ARG2,ARG3
1419 MSSVC C$SVEC
1420 M$GNINS <ADD #10,SP>
1421 .ENDM SETVEC
1422 .MACRO SLASH NUM
1423 .IF B NUM
1424 T$TEMP=1
1425 .IFF

```

1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482

```

T$TEMP=NUM
.ENDC
.REPT T$TEMP
  .IF GE SVCTAG
  .LIST
  ::////////////////////////////////////
  .NLIST
  .ENDC
  .ENDR
  .ENDM SLASH NUM
  .MACRO STARS
  .IF B NUM
  T$TEMP=1
  .IFF
  T$TEMP=NUM
  .ENDC
  .REPT T$TEMP
  .IF GE SVCTAG
  .LIST
  ::*****
  .NLIST
  .ENDC
  .ENDR
  .ENDM STARS
  .MACRO SVC
  CSREVISION=3
  CS$EDIT=3
  SVCINS=-1
  SVCTST=-1
  SVCSUB=-1
  SVCGBL=-1
  SVCTAG=-1
  .RADIX 10
  CSRESERV=0
  C$SETST=1
  CS$SUB=2
  C$ESUB=3
  CS$BSEG=4
  C$ESEG=5
  CS$CLP1=6
  CS$RDBU=7
  C$ESCAPE=8
  CS$INIT=9
  CS$CLEAN=10
  CSTPRI=11
  CSPNTB=12
  CSPNTX=13
  CSPNTS=14
  CSPNTF=15
  CS$INLP=16
  CS$RFLA=17
  CS$BRK=18
  CS$MSG=19
  CS$DRPT=20
  CS$RPT=21
  CS$GETB=22
  CS$GETW=23

```


1483	CSGPLO=24
1484	CSMEM=25
1485	CSEXIT=26
1486	CSRESET=27
1487	CSOPEN=28
1488	CSCLOS=29
1489	CSCVEC=30
1490	CSSVEC=31
1491	CSGPRI=32
1492	CSSPRI=33
1493	CSGPHRD=34
1494	CSGMAN=35
1495	CSDCLN=36
1496	CSCEFG=37
1497	CSSEFG=38
1498	CSREFG=39
1499	CSMANI=40
1500	CSDODU=41
1501	CSAU=42
1502	CSDU=43
1503	CSERSF=44
1504	CSERDF=45
1505	CSERHRD=46
1506	CSERSOFT=47
1507	CSERROR=48
1508	CSAUTO=49
1509	CSCCLK=50
1510	CSQIO=255
1511	.RADIX
1512	ESLOAD=35
1513	ESEND=2100
1514	TSTESTNUM=0
1515	TSSUBNUM=0
1516	TSERRNUM=0
1517	TSTAGNUM=10000
1518	TLSYM=10000
1519	TSTAGLEV=-1
1520	T\$NESTLEV=-1
1521	TSSAVLEV=-1
1522	TSSEGLEV=-1
1523	TSTSTMACH=-1
1524	TSTSTSEQ=0
1525	TSPPTNUM=0
1526	TSLAST=0
1527	TSGMANID=0
1528	FSMOD=0
1529	FSTEST=1
1530	FSSUB=2
1531	FSSEG=3
1532	FSHARD=4
1533	FSSOFT=5
1534	FSINIT=6
1535	FSCLEAN=7
1536	FSSRV=10
1537	FMSMSG=11
1538	FSRPT=12
1539	FSHW=13

8

1540	FSSW=14
1541	FSAU=15
1542	FSDU=16
1543	FSPWR=17
1544	FSAUTO=20
1545	FSPROT=21
1546	FSBGN=40
1547	FSEND=41
1548	FSJMP=50
1549	ISMOD=FSEND
1550	ISTST=FSEND
1551	ISSUB=FSEND
1552	ISSEG=FSEND
1553	ISINIT=FSEND
1554	ISCLN=FSEND
1555	ISSRV=FSEND
1556	ISMSG=FSEND
1557	ISRPT=FSEND
1558	ISAU=FSEND
1559	ISDU=FSEND
1560	ISPWR=FSEND
1561	ISSETUP=FSEND
1562	ISPTAB=FSEND
1563	ISAUTO=FSEND
1564	ISPROT=FSEND
1565	GSPRML=0
1566	GSPRMA=1
1567	GSPRMD=2
1568	GSDISP=3
1569	G\$XFER=4
1570	GSNO=0*D8
1571	G\$YES=1*D8
1572	GSRADB=0*D16
1573	GSRADO=1*D16
1574	GSRADD=2*D16
1575	GSRADL=5*D16
1576	GSRADA=6*D16
1577	G\$OFFSIZE=D254
1578	G\$DELM=D250
1579	G\$OFFSET=1*D256
1580	X\$OFFSET=1*D256
1581	G\$CNTOP=1*D128
1582	G\$EXCP=1*D256
1583	G\$LOLIM=1
1584	G\$HILIM=2
1585	X\$ALWAYS=0*D16
1586	X\$TRUE=1*D16
1587	X\$FALSE=2*D16
1588	OSBGNSFT=0
1589	OSBGNRPT=0
1590	OSGNSW=0
1591	OSAPTS=0
1592	OSAU=0
1593	OSDU=0
1594	OSERRTBL=0
1595	OSSETUP=0
1596	OS\$POINTER=0


```

1597      SLSYM=TSLSYM
1598      JSJMP=167
1599      .IIF NDF ASSEMBLY,          ASSEMBLY=10
1600      .IIF NDF DIAGMCALLS,      DIAGMCALLS=0
1601      .IF EQ DIAGMCALLS
1602      .MCALL MSMCHIGH
1603      MSMCHIGH
1604      .ENDC
1605      .IF EQ ASSEMBLY-10
1606      .MCALL MSMCLOW
1607      MSMCLOW
1608      .ENDC
1609      .ENDM      SVC
1610      .MACRO    XFER      $TAG
1611      .MCALL    MSXFER
1612      MSXFER    XSALWAYS,$TAG
1613      .ENDM      XFER
1614      .MACRO    XFERF     $TAG
1615      .MCALL    MSXFER
1616      MSXFER    X$FALSE,$TAG
1617      .ENDM      XFERF
1618      .MACRO    XFERT     $TAG
1619      .MCALL    MSXFER
1620      MSXFER    X$TRUE,$TAG
1621      .ENDM      XFERT
1622      .MACRO    MSMCLOW
1623      .MCALL    MSPUSH,MSPRIN,MSERRI,MSMASK,MSMSK1,MSXFER,MSIOSE,MSSTAR,MSHAPT,MSHNAP,MSDATA,MSGETT,M
1624      .MCALL    MSGNGB,MSGNLS,M$STL,M$TLAB,M$SVC,MSGENB,MSLDRO,MSRBRO,MSRNRO,M$PUT,M$PUT1,M$WORD,M$B
1625      .MCALL    M$EXSE,M$INCR,M$DECR,M$SETS,M$GETS,MSGEN,MSGNIN,M$COUN,M$SENDE,MSRADI,M$EXCP,M$CHEC,M$
1626      .MACRO    MSMCLOW
1627      .ENDM
1628      .ENDM      MSMCLOW
1629      .MACRO    MSPUSH  STACK,LEVEL,VALUE
1630      .MCALL    M$INCR,M$SETS
1631      M$INCR    LEVEL
1632      M$SETS    STACK,\LEVEL,VALUE
1633      .ENDM      MSPUSH
1634      .MACRO    M$PRINT CODE,FORMAT,P1,P2,P3,P4,P5,P6,P7,P8
1635      .MCALL    M$COUNT,M$PUT,M$GNINS,M$SVC
1636      TSARGCNT=1
1637      M$COUNT TSARGCNT,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1638      .IRP      N,<\TSARGCNT>
1639      M$PUT     #N,FORMAT,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
1640      .ENDM
1641      M$GNINS   <MOV      SP,R0>
1642      M$SVC     CODE
1643      .IRP      N,<\TSARGCNT*2+2>
1644      M$GNINS   <ADD      #N,SP>
1645      .ENDM
1646      .ENDM      M$PRINT
1647      .MACRO    M$ERRINS          NUMBER,ADDRESS,POINTER,CODE
1648      .MCALL    M$GNINS,M$WORD,M$STLAB
1649      M$SVC     CODE
1650      .RADIX    10
1651      .IF NB    NUMBER
1652      TSERRNUM=NUMBER
1653      .IFF

```

```
1654 .ERROR :MISSING ERROR-NR
1655 .ENDC
1656 M$WORD \T$ERRNUM
1657 .RADIX 8
1658 .IF NB ADDRESS
1659 M$WORD ADDRESS
1660 .IFF
1661 M$WORD 0
1662 .ENDC
1663 .IF NB POINTER
1664 M$WORD POINTER
1665 .IFF
1666 M$WORD 0
1667 .ENDC
1668 .ENDM MSERRINS
1669 .MACRO M$MASK P1,P2,P3,P4,P5
1670 .M$CALL M$WORD
1671 .IRP NR,<P1,P2,P3,P4,P5>
1672 M$WORD NR
1673 .ENDM
1674 .ENDM M$MASK
1675 .MACRO M$MSK1 P1,P2,P3,P4,P5
1676 .M$CALL M$WORD
1677 .NARG NR
1678 M$WORD \NR
1679 .ENDM M$MSK1
1680 .MACRO M$XFER $XFER,$ADDRESS
1681 .M$CALL M$GETTOP,M$WORD
1682 M$GETTOP T$NS,T$NESTLEV,T$CODE
1683 .IF NE F$HARD - T$CODE
1684 .IF NE F$SOFT - T$CODE
1685 .ERROR ;'XFER'S' MUST BE IN GPRM CODING.
1686 .MEXIT
1687 .ENDC
1688 .ENDC
1689 T$CODE=$ADDRESS - .
1690 .IF GT T$CODE-376
1691 .ERROR ;'XFER' RANGE TOO BIG
1692 .MEXIT
1693 .ENDC
1694 T$CODE=T$CODE * X$OFFSET
1695 T$CODE=T$CODE + G$XFER + $XFER
1696 M$WORD T$CODE
1697 .ENDM M$XFER
1698 .MACRO M$IOSET P1,P2,P3,P4,P5,P6
1699 .M$CALL M$WORD
1700 .IF NB P1
1701 M$WORD P1
1702 .IFF
1703 M$WORD 0
1704 .ENDC
1705 .IF NB P2
1706 M$WORD P2
1707 .IFF
1708 M$WORD 0
1709 .ENDC
1710 .IF NB P3
```



```

1711 MSWORD P3
1712 .IFF
1713 MSWORD 0
1714 .ENDC
1715 .IF NB P4
1716 MSWORD P4
1717 .IFF
1718 MSWORD 0
1719 .ENDC
1720 .IF NB P5
1721 MSWORD P5
1722 .IFF
1723 MSWORD 0
1724 .ENDC
1725 .IF NB P6
1726 MSWORD P6
1727 .IFF
1728 MSWORD 0
1729 .ENDC
1730 .ENDM MSIOSET
1731 .MACRO MSSTART ERR,?TAG
1732 .MCALL MSGNINS,MSGNTAG,BNERROR
1733 BNERROR TAG
1734 MSGNINS <JSR PC,ERR>
1735 MSGNTAG TAB
1736 .ENDM MSSTART
1737 .MACRO MSHAPT FILNAM,REVLEV,DEPO,LONGST,DIAGTYPE,DIAGPRO
1738 .MCALL MSGNGBL,MSGNINS,MSBYTE,MSDATA
1739 .IF B FILNAM
1740 .ERROR ; MISSING DIAG. NAME
1741 .MEXIT
1742 .ENDC
1743 MSGNGBL LSNAME,<<;DIAGNOSTIC NAME>>
1744 TSARGCNT=^D7
1745 .IRPC CHAR,<FILNAM>
1746 MSGNINS <.ASCII /CHAR/>
1747 TSARGCNT=TSARGCNT-1
1748 .IF LT TSARGCNT
1749 .ERROR ;PROG NAME TOO BIG
1750 .MEXIT
1751 .ENDC
1752 .ENDM
1753 .REPT TSARGCNT
1754 MSBYTE 0
1755 .ENDR
1756 MSBYTE 0
1757 .IF NB REVLEV
1758 MSDATA LSREV,..ASCII,</REVLEV/>,<<;REVISION LEVEL>>
1759 .IFF
1760 .ERROR ; MISSING REV. LEVEL
1761 MSDATA LSREV,..ASCII,</?/?/>,<<;REVISION LEVEL>>
1762 .ENDC
1763 .IF NB DEPO
1764 MSDATA LSDEPO,..ASCII,</DEPO/>,<<;DEPO>>
1765 .IFF
1766 .ERROR ; MISSING DEPO.
1767 MSDATA LSDEPO,..ASCII,</?/?/>,<<;DEPO>>

```

```

1768 .ENDC
1769 .IF EQ OSSETUP
1770 MSDATA LSUNIT,.WORD,0,<<;NUMBER OF UNITS>>
1771 .IFF
1772 MSDATA LSUNIT,.WORD,TSPTHV,<<;NUMBER OF UNITS>>
1773 .ENDC
1774 .IF B LONGST
1775 .ERROR ; LONGEST TEST TIME MISSING
1776 MSDATA LSTIML,.WORD,0,<<;LONGEST TEST TIME>>
1777 .IFF
1778 MSDATA LSTIML,.WORD,LONGST,<<;LONGEST TEST TIME>>
1779 .ENDC
1780 MSDATA LSHPCP,.WORD,LSHARD,<<;POINTER TO H.W. QUES.>>
1781 .IF EQ OSBGNSFT
1782 MSDATA LSSPCP,.WORD,0,<<;POINTER TO S.W. QUES.>>
1783 .IFF
1784 MSDATA LSSPCP,.WORD,LSSOFT,<<;POINTER TO S.W. QUES.>>
1785 .ENDC
1786 MSDATA LSHPTP,.WORD,LSHW,<<;PTR. TO DEF. H.W. PTABLE>>
1787 .IF EQ OSGNSW
1788 MSDATA LSSPTP,.WORD,0,<<;PTR. TO S.W. PTABLE>>
1789 .IFF
1790 MSDATA LSSPTP,.WORD,LSSW,<<;PTR. TO S.W. PTABLE>>
1791 .ENDC
1792 MSDATA L$LADP,.WORD,L$LAST,<<;DIAG. END ADDRESS>>
1793 .IF EQ OSAPTS
1794 MSDATA L$STA,.WORD,0,<<;RESERVED FOR APT STATS>>
1795 .IFF
1796 MSDATA L$STA,.WORD,0,<<;RESERVED FOR APT STATS>>
1797 .ENDC
1798 MSDATA LSCO,.WORD,0
1799 .IF B DIAGTYPE
1800 .ERROR ; MISSING DIAG. TYPE
1801 MSDATA L$DTYP,.WORD,0,<<;DIAGNOSTIC TYPE>>
1802 .IFF
1803 .IF IDN 0,DIAGTYPE
1804 MSDATA L$DTYP,.WORD,DIAGTYPE,<<;DIAGNOSTIC TYPE>>
1805 .IFF
1806 .IF IDN 1,DIAGTYPE
1807 MSDATA L$DTYP,.WORD,DIAGTYPE,<<;DIAGNOSTIC TYPE>>
1808 .IFF
1809 MSDATA L$DTYP,.WORD,0,<<;DIAGNOSTIC TYPE>>
1810 .ERROR ; INVALID DIAG. TYPE (0=DIAG, 1=EXERCISER)
1811 .ENDC
1812 .ENDC
1813 .ENDC
1814 MSDATA LSAPT,.WORD,0,<<;APT EXPANSION>>
1815 MSDATA LSDTP,.WORD,L$DISPATCH,<<;PTR. TO DISPATCH TABLE>>
1816 .IF B DIAGPRIO
1817 MSDATA LSPRIO,.WORD,0,<<;DIAGNOSTIC RUN PRIORITY>>
1818 .IFF
1819 MSDATA LSPRIO,.WORD,DIAGPRIO,<<;DIAGNOSTIC RUN PRIORITY>>
1820 .ENDC
1821 .ENDM MSHAPT
1822 .MACRO MSHNAP
1823 .MCALL MSDATA,MSWORD,MSGNINS
1824 MSDATA L$ENVI,.WORD,0,<<;FLAGS DESCRIBE HOW IT WAS SETUP>>

```



```

1825 MSDATA LSEXP1,.WORD,0,<<;EXPANSION WORD>>
1826 MSDATA LSMREV,.BYTE,C$REVISION,<<;SVC REV AND EDIT #>>
1827 MSGNINS <.BYTE C$EDIT>
1828 MSDATA LSEF,.WORD,0 ,<<;DIAG. EVENT FLAGS>>
1829 MSWORD 0
1830 MSDATA LSSPC,.WORD,0
1831 MSDATA LSDEVP,.WORD,LSDVTYP ,<<; POINTER TO DEVICE TYPE LIST>>
1832 .IF EQ OSBGNRPT
1833 MSDATA LSREPP,.WORD,0,<<;PTR. TO REPORT CODE>>
1834 .IFF
1835 MSDATA LSREPP,.WORD,LSRPT,<<;PTR. TO REPORT CODE>>
1836 .ENDC
1837 MSDATA LSEXP4,.WORD,0
1838 MSDATA LSEXP5,.WORD,0
1839 .IF EQ OSAU
1840 MSDATA LSAUT,.WORD,0,<<;PTR. TO ADD UNIT CODE>>
1841 .IFF
1842 MSDATA LSAUT,.WORD,LSAU,<<;PTR. TO ADD UNIT CODE>>
1843 .ENDC
1844 .IF EQ OSDU
1845 MSDATA LSDUT,.WORD,0,<<;PTR. TO DROP UNIT CODE>>
1846 .IFF
1847 MSDATA LSDUT,.WORD,LSDU,<<;PTR. TO DROP UNIT CODE>>
1848 .ENDC
1849 MSDATA LSLUN,.WORD,0 ,<<;LUN FOR EXERCISERS TO FILL>>
1850 MSDATA LSDESP,.WORD,LSDESC ,<<;POINTER TO DIAG. DESCRIPTION>>
1851 MSDATA L$LOAD,EMT,E$LOAD ,<<;GENERATE SPECIAL AUTOLOAD EMT>>
1852 .IF EQ OSERRTBL
1853 MSDATA L$ETP,.WORD,0,<<;POINTER TO ERRTBL>>
1854 .IFF
1855 MSDATA L$ETP,.WORD,LSERRTBL,<<;POINTER TO ERRTBL>>
1856 .ENDC
1857 MSDATA LSICP,.WORD,LSINIT,<<;PTR. TO INIT CODE>>
1858 MSDATA LSCCP,.WORD,LSCLEAN,<<;PTR. TO CLEAN-UP CODE>>
1859 MSDATA LSACP,.WORD,LSAUTO,<<;PTR. TO AUTO CODE>>
1860 MSDATA LSPRT,.WORD,LSPROT,<<;PTR. TO PROTECT TABLE>>
1861 MSDATA L$TEST,.WORD,0,<<;TEST NUMBER>>
1862 MSDATA L$DLY,.WORD,0,<<;DELAY COUNT>>
1863 MSDATA L$HIME,.WORD,0,<<;PTR. TO HIGH MEM>>
1864 .ENDM MSHNAP
1865 .MACRO MSDATA LABEL,DTYPE,DATA,COMMENT
1866 .MCALL MSGNGBL,MSGNINS
1867 .IF NB COMMENT
1868 MSGNGBL LABEL,<COMMENT>
1869 .ENDC
1870 .IF B COMMENT
1871 MSGNGBL LABEL
1872 .ENDC
1873 MSGNINS <DTYPE DATA>
1874 .ENDM MSDATA
1875 .MACRO MSGETTOP STACK,LEVEL,VALUE
1876 .MCALL MSGETS
1877 MSGETS STACK,\LEVEL,VALUE
1878 .ENDM MSGETTOP
1879 .MACRO MSPOP STACK,LEVEL,VALUE
1880 .MCALL MSGETS,MSDECR
1881 .IF LT LEVEL

```

```

1882 .ERROR LEVEL ; MACRO STACK UNDERFLOW
1883 .MEXIT
1884 .ENDC
1885 MSGETS STACK,\LEVEL,VALUE
1886 MSDECR LEVEL
1887 .ENDM MSPOP
1888 .MACRO MSGNTAG SP,$T
1889 .MCALL MSGEN
1890 MSGEN SP,\$T,SVCTAG
1891 .ENDM MSGNTAG
1892 .MACRO MSGNTEST $TN
1893 .MCALL MSGEN
1894 MSGEN T,'$TN',,SVCTST
1895 .ENDM MSGNTEST
1896 .MACRO MSGNSUB $TN,$SN
1897 .MCALL MSGEN
1898 MSGEN T'$TN',, '$SN',SVCSUB
1899 .ENDM MSGNSUB
1900 .MACRO MSGNGBL TAG,COMMENT
1901 .MCALL MSGEN
1902 .IF NB,TAG
1903 .IF NB,COMMENT
1904 MSGEN TAG,.,SVCGBL,COMMENT
1905 .ENDC
1906 .IF B,COMMENT
1907 MSGEN TAG,.,SVCGBL,< >
1908 .ENDC
1909 .ENDC
1910 .ENDM MSGNGBL
1911 .MACRO MSGNLS ARG
1912 .MCALL MSGEN
1913 .IF NB ARG
1914 MSGEN \ARG,$,SVCTAG
1915 .IFF
1916 MSGEN \$$LSYM,$,SVCTAG
1917 .ENDC
1918 .ENDM MSGNLS
1919 .MACRO MSTSTLAB
1920 .MCALL MSTLABEL,MSINCR
1921 .IIF LT TSTSTMAC, .MEXIT
1922 .RADIX 10
1923 MSTLABEL \TSTESTNUN,\TSTSTSEQ
1924 .RADIX 8
1925 MSINCR TSTSTSEQ
1926 .ENDM MSTSTLAB
1927 .MACRO MSTLABEL TST,SEQ
1928 .MCALL MSGEN
1929 MSGEN S'TST,S'SEQ,SVCTAG
1930 .ENDM MSTLABEL
1931 .MACRO MSSVC CODE
1932 .MCALL MSGNINS,MSTSTLAB
1933 MSTSTLAB
1934 MSGNINS <TRAP CODE>
1935 .ENDM MSSVC
1936 .MACRO MSGENBR $$BR,$$SYN,$$TAG
1937 .MCALL MSGNINS
1938 MSGNINS <$$BR $$SYN'$$TAG>

```



```

1939 .ENDM MSGENBR
1940 .MACRO MSLDRO ARG,NAME
1941 .MCALL MSGNINS
1942 .IF NB ARG
1943 .IF DIF R0,ARG
1944 MSGNINS <MOV ARG,R0>
1945 .ENDC
1946 .IFF
1947 .ERROR ;MISSING ARG. ON 'NAME'
1948 .ENDC
1949 .ENDM MSLDRO
1950 .MACRO MSRBRO ARG,NAME
1951 .MCALL MSGNINS
1952 .IF NB ARG
1953 .IF DIF R0,ARG
1954 MSGNINS <MOVB R0,ARG>
1955 .ENDC
1956 .IFF
1957 .ERROR ;MISSING ARG. ON 'NAME'
1958 .ENDC
1959 .ENDM MSRBRO
1960 .MACRO MSRNRO ARG,NAME
1961 .MCALL MSGNINS
1962 .IF NB ARG
1963 .IF DIF R0,ARG
1964 MSGNINS <MOV R0,ARG>
1965 .ENDC
1966 .IFF
1967 .ERROR ;MISSING ARG. ON 'NAME'
1968 .ENDC
1969 .ENDM MSRNRO
1970 .MACRO MSPUT P0,P1,P2,P3,P4,P5,P6,P7,P8,P9
1971 .MCALL MSPUT1
1972 .IF NB <P1>
1973 MSPUT <P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>,<P9>
1974 .ENDC
1975 .IF NB <P0>
1976 MSPUT1 P0
1977 .ENDC
1978 .ENDM MSPUT
1979 .MACRO MSPUT1 ARG1,ARG2
1980 .MCALL MSGNINS
1981 .IF B ARG2
1982 MSGNINS <MOV ARG1,-(SP)>
1983 .IFF
1984 .IF DIF B,<ARG1>
1985 .ERROR ;ILL. ARG. 'ARG1' ONLY 'B' IS ALLOWED.
1986 .MEXIT
1987 .ENDC
1988 MSGNINS <CLR -(SP)>
1989 MSGNINS <BISB ARG2,(SP)>
1990 .ENDC
1991 .ENDM MSPUT1
1992 .MACRO MSWORD ARG
1993 .MCALL MSGNINS
1994 .IRP N,<ARG>
1995 MSGNINS <.WORD N>

```

```

1996 .ENDM
1997 .ENDM MSWORD
1998 .MACRO MSBYTE ARG
1999 .MCALL MSGNINS
2000 .IRP N,<ARG>
2001 MSGNINS <.BYTE N>
2002 .ENDM
2003 .ENDM MSBYTE
2004 .MACRO M$ESCAPE TAG1
2005 .MCALL MSGNINS
2006 MSGNINS <.WORD L'TAG1-.>
2007 .ENDM M$ESCAPE
2008 .MACRO M$ESCSEG TAG1
2009 .MCALL MSGNINS
2010 MSGNINS <.WORD TAG1'$-.>
2011 .ENDM M$ESCSEG
2012 .MACRO M$EXIT TAG1
2013 .MCALL MSGNINS
2014 MSGNINS <.WORD L'TAG1-.>
2015 .ENDM M$EXIT
2016 .MACRO M$EXTJ TAG1
2017 .MCALL MSGNINS
2018 MSGNINS <.WORD L'TAG1-2-.>
2019 .ENDM M$EXTJ
2020 .MACRO M$EXSEG TAG1
2021 .MCALL MSGNINS
2022 MSGNINS <.WORD TAG1'$-.>
2023 .ENDM M$EXSEG
2024 .MACRO M$INCR VAL
2025 VAL=VAL+1
2026 .ENDM M$INCR
2027 .MACRO M$DECR VAL
2028 VAL=VAL-1
2029 .ENDM M$DECR
2030 .MACRO M$SETS $STACK,$LEVEL,$VALUE
2031 $STACK'$LEVEL=$VALUE
2032 .ENDM M$SETS
2033 .MACRO M$GETS $STACK,$LEVEL,$VALUE
2034 $VALUE=$STACK'$LEVEL
2035 .ENDM M$GETS
2036 .MACRO M$GEN $SP,$ST,LSTCNTRL,COMMENT
2037 .IF LE LSTCNTRL
2038 .IIF EQ LSTCNTRL,.LIST
2039 $SP'$ST: COMMENT
2040 .IIF EQ LSTCNTRL,.NLIST
2041 .MEXIT
2042 .ENDC
2043 .LIST
2044
2045 .NLIST
2046 .ENDM M$GEN
2047 .MACRO MSGNINS INSTR
2048 .IF LT SVCINS
2049 INSTR
2050 .MEXIT
2051 .ENDC
2052 .IF EQ SVCINS

```

\$SP'\$ST:


```

2053      .LIST
2054          INSTR
2055          .NLIST
2056      .MEXIT
2057      .ENDC
2058      .IF GT SVCINS
2059      .LIST
2060
2061          .NLIST
2062          INSTR
2063      .ENDC
2064      .ENDM MSGNINS
2065      .MACRO MSCOUNT COUNT,P0,P1,P2,P3,P4,P5,P6,P7,P8
2066      .IF NB <P1>
2067      MSCOUNT COUNT,<P1>,<P2>,<P3>,<P4>,<P5>,<P6>,<P7>,<P8>
2068      .ENDC
2069      .IF NB <P0>
2070      COUNT=COUNT+1
2071      .ENDC
2072      .ENDM MSCOUNT
2073      .MACRO MSENDERR          $NAME,$FLAG
2074      .ERROR ;'"$NAME"' STMT OUT OF SEQUENCE.
2075      .ENDM MSENDERR
2076      .MACRO MSRADIX $IN,$OUT
2077      .IF IDN B,$IN
2078      $OUT=GSRADB
2079      .MEXIT
2080      .ENDC
2081      .IF IDN O,$IN
2082      $OUT=GSRADO
2083      .MEXIT
2084      .ENDC
2085      .IF IDN D,$IN
2086      $OUT=GSRADD
2087      .MEXIT
2088      .ENDC
2089      .IF IDN L,$IN
2090      $OUT=GSRADL
2091      .MEXIT
2092      .ENDC
2093      .IF IDN A,$IN
2094      $OUT=GSRADA
2095      .MEXIT
2096      .ENDC
2097      $OUT=-1
2098      .ERROR ;ILL. RADIX '$IN'
2099      .ENDM MSRADIX
2100      .MACRO MSEXCP WORD1,EXWD,EXCD,LIMIT,LIM1,LIM2
2101      .IF IDN <@>,<LIM1>
2102      .IF LT G$OFSIZE-LIM2
2103      .ERROR ;INDIRECT PAR. TOO BIG
2104      .MEXIT
2105      .ENDC
2106      LIMIT=LIM2/2
2107      WORD1=WORD1 ! G$EXCP
2108      EXWD=EXWD ! EXCD
2109      .IFF
2110      .IF B,LIM2

```

```
2110      LIMIT=LIM1
2111      .IFF
2112      .ERROR ;ILL. DEFERRED MODE
2113      .ENDC
2114      .ENDC
2115      .ENDM MSEXCP
2116      .MACRO MSCHECK NAME,TAG,FLAG
2117      .IF IDN <TST>,<NAME>
2118      TAG=T$$TEST
2119      FLAG=ISTST
2120      .MEXIT
2121      .ENDC
2122      .IF IDN <SUB>,<NAME>
2123      TAG=T$$SUB
2124      FLAG=ISSUB
2125      .MEXIT
2126      .ENDC
2127      .IF IDN <SEG>,<NAME>
2128      TAG=T$$SEG
2129      FLAG=ISSEG
2130      .MEXIT
2131      .ENDC
2132      .IF IDN <INIT>,<NAME>
2133      TAG=T$$INIT
2134      FLAG=ISINIT
2135      .MEXIT
2136      .ENDC
2137      .IF IDN <CLN>,<NAME>
2138      TAG=T$$CLEAN
2139      FLAG=ISCLN
2140      .MEXIT
2141      .ENDC
2142      .IF IDN <SRV>,<NAME>
2143      TAG=T$$SRV
2144      FLAG=ISSRV
2145      .MEXIT
2146      .ENDC
2147      .IF IDN <MSG>,<NAME>
2148      TAG=T$$MSG
2149      FLAG=FSJMP
2150      .MEXIT
2151      .ENDC
2152      .IF IDN <RPT>,<NAME>
2153      TAG=T$$RPT
2154      FLAG=FSJMP
2155      .MEXIT
2156      .ENDC
2157      .IF IDN <DU>,<NAME>
2158      TAG=T$$DU
2159      FLAG=FSJMP
2160      .MEXIT
2161      .ENDC
2162      .IF IDN <AU>,<NAME>
2163      TAG=T$$AU
2164      FLAG=FSJMP
2165      .MEXIT
2166      .ENDC
```



```
2167 .ENDM MSCHECK
2168 .MACRO MSDEFAULT $IN,$OUT
2169 .IF IDN YES,$IN
2170 $OUT=G$YES
2171 .MEXIT
2172 .ENDC
2173 .IF IDN NO,$IN
2174 $OUT=G$NO
2175 .MEXIT
2176 .ENDC
2177 $OUT=-1
2178 .ERROR ;DEFAULT '$IN' MUST BE 'YES' OR 'NO'
2179 .ENDM MSDEFAULT
2180 .MACRO MSCNTOP $IN,WORD1
2181 .IF NB $IN
2182 .IF NE ISSFT - F$BGN
2183 .ERROR ; 'COUNT' OPTION VALID ONLY IN S.W. QUES.
2184 .MEXIT
2185 .ENDC
2186 WORD1=WORD1 ! G$CNTOP
2187 .ENDC
2188 .ENDM MSCNTOP
```

1		.TITLE CZRLGCO RL11/RLV11 CTLR TST 1
2		.ENABLE AMA
3	000000	.ENABLE ABS
4		.NLIST ME,CND,MD
5		.MCALL SVC
6		
7	000000	SVC
8		SVCINS=0
9	000000	SVCTAG=0
10	002000	.=2000
11		
12		
13	002000	POINTER BGNSFT,BGNSW,BGNDU,BGNAU
14		
15	002000	BGNMOD MDHEDR
16		
17	002000	HEADER CZRLG,C,0,7,0
	002000	.ASCII /C/
	002001	.ASCII /Z/
	002002	.ASCII /R/
	002003	.ASCII /L/
	002004	.ASCII /G/
	002005	.BYTE 0
	002006	.BYTE 0
	002007	.BYTE 0
	002010	.ASCII /C/
	002011	.ASCII /O/
	002012	.WORD 0
	002014	.WORD 7
	002016	.WORD LSHARD
	002020	.WORD LSSOFT
	002022	.WORD LSHW
	002024	.WORD LSSW
	002026	.WORD L\$LAST
	002030	.WORD 0
	002032	.WORD 0
	002034	.WORD 0
	002036	.WORD 0
	002040	.WORD L\$DISPATCH
	002042	.WORD 0
	002044	.WORD 0
	002046	.WORD 0
	002050	.BYTE C\$REVISION
	002051	.BYTE C\$EDIT
	002052	.WORD 0
	002054	.WORD 0
	002056	.WORD 0
	002060	.WORD L\$DVTYP
	002062	.WORD 0
	002064	.WORD 0
	002066	.WORD 0
	002070	.WORD L\$AU
	002072	.WORD L\$DU
	002074	.WORD 0
	002076	.WORD L\$DESC
	002100	EMT E\$LOAD
	002102	.WORD 0

002104 011772
 002106 012756
 002110 012534
 002112 011764
 002114 000000
 002116 000000
 002120 000000

.WORD LSINIT
 .WORD LSCLEAN
 .WORD LSAUTO
 .WORD LSPROT
 .WORD 0
 .WORD 0
 .WORD 0

18
 19 002122

ENDMOD

20
 21 002122

DESCRIPT <CZRLG TESTS CONTROLLER FUNCTIONS, INTERFACE LOGIC, REGISTER OPERATION>
 .ASCIZ /CZRLG TESTS CONTROLLER FUNCTIONS, INTERFACE LOGIC, REGISTER OPERATION/

002122	103	132	122
002125	114	107	040
002130	124	105	123
002133	124	123	040
002136	103	117	116
002141	124	122	117
002144	114	114	105
002147	122	040	106
002152	125	116	103
002155	124	111	117
002160	116	123	054
002163	040	111	116
002166	124	105	122
002171	106	101	103
002174	105	040	114
002177	117	107	111
002202	103	054	040
002205	122	105	107
002210	111	123	124
002213	105	122	040
002216	117	120	105
002221	122	101	124
002224	111	117	116
002227	000		

22 002230			
002230	122	114	060
002233	061	054	122
002236	114	060	062
002241	000		

DEV TYP .EVEN
 <RL01,RL02>
 .ASCIZ /RL01,RL02/

23
 24 002242
 25
 26 002242

.EVEN
 BGNMOD GLBEQAT
 EQUALS

100000
 040000
 020000
 010000
 004000
 002000
 001000
 000400

... BIT DIFINITIONS
 ...
 BIT15== 100000
 BIT14== 40000
 BIT13== 20000
 BIT12== 10000
 BIT11== 4000
 BIT10== 2000
 BIT09== 1000
 BIT08== 400

000200 BIT07== 200
 000100 BIT06== 100
 000040 BIT05== 40
 000020 BIT04== 20
 000010 BIT03== 10
 000004 BIT02== 4
 000002 BIT01== 2
 000001 BIT00== 1

001000 BIT9== BIT09
 000400 BIT8== BIT08
 000200 BIT7== BIT07
 000100 BIT6== BIT06
 000040 BIT5== BIT05
 000020 BIT4== BIT04
 000010 BIT3== BIT03
 000004 BIT2== BIT02
 000002 BIT1== BIT01
 000001 BIT0== BIT00

EVENT FLAG DEFINITIONS
 EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION

000040 EF.START== 32. : START COMMAND WAS ISSUED
 000037 EF.RESTART== 31. : RESTART COMMAND WAS ISSUED
 000036 EF.CONTINUE== 30. : CONTINUE COMMAND WAS ISSUED
 000035 EF.NEW== 29. : A NEW PASS HAS BEEN STARTED
 000034 EF.PWR== 28. : A POWER-FAIL/POWER-UP OCCURRED

PRIORITY LEVEL DEFINITIONS

000340 PRI07== 340
 000300 PRI06== 300
 000240 PRI05== 240
 000200 PRI04== 200
 000140 PRI03== 140
 000100 PRI02== 100
 000040 PRI01== 40
 000000 PRI00== 0

OPERATOR FLAG BITS

000004 EVL== 4
 000010 LOT== 10
 000020 ADR== 20
 000040 IDU== 40
 000100 ISR== 100
 000200 UAM== 200
 000400 BOE== 400
 001000 PNT== 1000
 002000 PRI== 2000
 004000 IXE== 4000
 010000 IBE== 10000
 020000 IER== 20000
 040000 LOE== 40000
 100000 HOE== 100000

27	000001	DRDY=BIT0	:DRIVE READY (RLCS)
28	000100	INTEN=BIT6	: INTERRUPT ENABLE (RLCS)
29	100000	ERR=BIT15	: RL11 ERROR (RLCS)
30	040000	DERR=BIT14	: RLO1 DRIVE ERROR (RLCS)
31	002000	OPI=BIT10	: OPERATION INCOMPLETE (RLCS)
32	000200	CRDY=BIT7	: CONTROLLER READY (RLCS)
33	000040	BA17=BIT5	: EXTENDED ADDRESS BIT 17 (RLCS)
34	000020	BA16=BIT4	: EXTENDED ADDRESS BIT 16 (RLCS)
35	020000	NXM=BIT13	: NON-EXISTANT MEMORY (RLCS)
36	000000	DS0=0	: DRIVE SELECT 0 (RLCS)
37	000400	DS1=BIT8	: DRIVE SELECT 1 (RLCS)
38	001000	DS2=BIT9	: DRIVE SELECT 2 (RLCS)
39	001400	DS3=BIT8!BIT9	: DRIVE SELECT 3 (RLCS)
40	000000	NOOP0=0	: FUNCTION-NOOP(0)
41	000016	NOOP7=BIT1!BIT2!BIT3	: FUNCTION-NOOP(7)
42	000002	WRCHK=BIT1	: WRITE CHECK FUNCTION
43	000004	GSTAT=BIT2	: GET STATUS FUNCTION
44	000006	SEEK=BIT2!BIT1	: SEEK FUNCTION
45	000010	RDHDR=BIT3	: READ HEADER FUNCTION
46	000012	WRITE=BIT3!BIT1	: WRITE DATA FUNCTION
47	000014	READ=BIT3!BIT2	: READ DATA FUNCTION
48	000202	GODRVR=BIT1!BIT7	: CRDY AND DRDY
49	000010	DRST=BIT3	: DRIVE RESET (RLDA)
50	000002	GSBIT=BIT1	: GET STATUS BIT (RLDA)
51	000001	MK=BIT0	: MARKER BIT (RLDA)
52	000004	SIGN=BIT2	: SIGN BIT (RLDA)
53	000100	RHMS=BIT6	: HEAD SELECT IN READ HEADER
54	000100	STHS=BIT6	: HEAD SELECT IN STATUS BACK
55	000020	DAHS=BIT4	: HEAD SELECT IN SEEK
56			
57		:OFFSET FOR HARDWARE P-TABLE	
58			
59	000000	CSR=0	
60	000002	VECT=2	
61	000004	PRIOR=4	
62	000006	TYPDR=6	
63	000010	DRBT=10	
64	000012	CNT=12	
65			
66		:OFFSET FOR SOFTWARE P-TABLE	
67			
68	000000	DLT=0	
69	000002	ELT=2	
70	000004	SIZE=4	
71			
72	002242	ENDMOD	
73			
74	002242	BGNMOD GLBDAT	
75			
76		.SBTTL GLOBAL DATA	
77			
78	002242	PWRFLG: .WORD 0	
79	002244	UUT: .WORD 0	
80	002246	UNITST: .WORD 0	
81	002250	RLCS: .WORD 0	: LOGICAL ADDRESS OF CS
82	002252	RLBA: .WORD 0	: LOGICAL ADDRESS OF BA
83	002254	RLDA: .WORD 0	: LOGICAL ADDRESS OF DA

GLOBAL DATA

84	002256	000000	RLMP:	.WORD	0	:LOGICAL ADDRESS OF MP
85	002260	000000	RLBE:	.WORD	0	:LOGICAL ADDRESS OF BE
86	002262	000000	BCSR:	.WORD	0	
87	002264	000000	BPRIOR:	.WORD	0	
88	002266	000000	BVEC:	.WORD	0	
89	002270	000000	DRIVE:	.WORD	0	:DRIVE UNDER TEST
90	002272	000000	B.CS:	.WORD	0	:CS - BEFORE OPERATION
91	002274	000000	B.BA:	.WORD	0	:BA - BEFORE OPERATION
92	002276	000000	B.DA:	.WORD	0	:DA - BEFORE OPERATION
93	002300	000000	B.MP:	.WORD	0	:MP - BEFORE OPERATION
94	002302	000000	B.BE:	.WORD	0	:BE - BEFORE OPERATION
95	002304	000000	DERFLG:	.WORD		
96	002306	000000	E.CS:	.WORD	0	:CS - AT OCCURANCE OF ERROR
97	002310	000000	E.BA:	.WORD	0	:BA - AT OCCRUANCE OF ERROR
98	002312	000000	E.DA:	.WORD	0	:DA - AT OCCURANCE OF ERROR
99	002314	000000	E.MP:	.WORD	0	:MP - AT OCCURANCE OF ERROR
100	002316	000000	E.MP1:	.WORD	0	
101	002320	000000	E.MP2:	.WORD	0	:MP - AT OCCURANCE OF ERROR READ HEADER
102	002322	000000	E.BE:	.WORD	0	:BE - AT OCCURANCE OF ERROR RLV12 ONLY
103	002324	000000	PFLG:	.WORD	0	:PROCESSOR TYPE, 0=UNIBUS, 1=Q-BUS
104	002326	000000	TRPFLG:	.WORD	0	
105	002330	000000	INTFLG:	.WORD	0	:INTERRUPT OCCURRENCE FLAG
106	002332	000000	LDCSR:	.WORD	0	:LOCATION TO FORM RLCS
107	002334	000077	SECMSK:	.WORD	77	:MASK OUT SECTOR
108	002336	120001	XPOLY:	.WORD	120001	:POLYNOMIAL FOR CRC 16
109	002340	000004	ERRVEC:	.WORD	4	
110	002342	000000	BCCFBK:	.WORD	0	:LOCATION USED BY "SIMBCC"
111	002344	000000	CALBCC:	.WORD	0	:LOCATION USED BY "SIMBCC"
112	002346	000000	TEMP2:	.WORD	0	:LOCATION USED BY "SIMBCC"
113	002350	000000	TEMP3:	.WORD	0	:LOCATION USED BY "SIMBCC"
114	002352	000000	TEMP4:	.WORD	0	:LOCATION USED BY "SIMBCC"
115	002354	000000	TMP0:	.WORD	0	
116	002356	000000	TMP1:	.WORD	0	
117	002360	000000	TMP2:	.WORD	0	
118	002362	000000	GDDAT:	.WORD	0	
119	002364	000000	BDDAT:	.WORD	0	
120	002366	000000	FIRST:	.WORD	0	:FIRST SECTOR READ
121	002370	177700	CYLMASK:	.WORD	177700	:MASK CYLINDER AND HEAD SELECT
122	002372	000050	MXSEC1:	.WORD	40.	:MAX SECTOR ADDRESS +1
123	002374	000047	MAXSEC:	.WORD	39.	:MAX SECTOR ADDRESS
124	002376	000000	DWORD:	.WORD	0	:DIFFERENCE WORD (SEEK)
125	002400	177600	MAXCYL:	.WORD	177600	:MAXIMUM CYLINDER ADDRESS
126	002402	000000	SVHD:	.WORD	0	:SAVE CURRENT HEAD SELECT
127	002404	000000	WHY:	.WORD	0	:REASON FOR DROP UNIT
128						
129	002406	000000	T.DRIVE:	.WORD	0	:DRIVE TYPE
130	002410	000000	T.CNTRLR:	.WORD	0	:CONTOLLER TYPE
131	002412	000000	TMPFNC:	.WORD	0	
132	002414	000000	DLYCNT:	.WORD	0	:DELAY COUNTER
133						
134						
135						
136						
137	002416	000000	BEGPAT:	0		:GROWING 1
138	002420	000001		1		
139	002422	000003		3		
140	002424	000007		7		

141	002426	000017	17
142	002430	000037	37
143	002432	000077	77
144	002434	000177	177
145	002436	000377	377
146	002440	000777	777
147	002442	001777	1777
148	002444	003777	3777
149	002446	007777	7777
150	002450	017777	17777
151	002452	037777	37777
152	002454	077777	77777
153	002456	177777	177777
154	002460	177776	177776
155	002462	177774	177774
156	002464	177770	177770
157	002466	177760	177760
158	002470	177740	177740
159	002472	177700	177700
160	002474	177600	177600
161	002476	177400	177400
162	002500	177000	177000
163	002502	176000	176000
164	002504	174000	174000
165	002506	170000	170000
166	002510	160000	160000
167	002512	140000	140000
168	002514	100000	100000
169			
170	002516	000000	000000
171	002520	000001	1
172	002522	000002	2
173	002524	000004	4
174	002526	000010	10
175	002530	000020	20
176	002532	000040	40
177	002534	000100	100
178	002536	000200	200
179	002540	000400	400
180	002542	001000	1000
181	002544	002000	2000
182	002546	004000	4000
183	002550	010000	10000
184	002552	020000	20000
185	002554	040000	40000
186	002556	100000	100000
187	002560	177777	177777
188	002562	177776	177776
189	002564	177775	177775
190	002566	177773	177773
191	002570	177767	177767
192	002572	177757	177757
193	002574	177737	177737
194	002576	177677	177677
195	002600	177577	177577
196	002602	177377	177377
197	002604	176777	176777

:GROWING 0

:WALKING 1

:WALKING 0

198 002606 175777
 199 002610 173777
 200 002612 167777
 201 002614 157777
 202 002616 137777
 203 002620 077777
 204 002622 177777
 205 002624 000000
 206
 207
 208
 209 002626 000200
 210 002630 000400
 211 002632 001000
 212 002634 002000
 213 002636 004000
 214 002640 010000
 215 002642 020000
 216 002644 040000
 217 002646 077600
 218 002650 077400
 219 002652 076600
 220 002654 075600
 221 002656 073600
 222 002660 067600
 223 002662 057600
 224 002664 037600
 225 002666 077600
 226 002670 000200
 227 002672 000600
 228 002674 001600
 229 002676 003600
 230 002700 007600
 231 002702 017600
 232 002704 037600
 233 002706 077600
 234 002710 077400
 235 002712 077000
 236 002714 076000
 237 002716 074000
 238 002720 070000
 239 002722 060000
 240 002724 040000
 241 002726 000000
 242 002730 100000
 243 002732 037600
 244 002734 077600
 245
 246 002736 177600
 247 002740 177400
 248 002742 176600
 249 002744 173600
 250 002746 167600
 251 002750 157600
 252 002752 137600
 253 002754 177000
 254 002756 176000

175777
 173777
 167777
 157777
 137777
 077777
 177777
 ENDPAT: 000000

.SBTTL PATTERNS FOR DIFFERENCE WORD

SKLST: .WORD BIT7
 .WORD BIT8 ;SHIFTING 1
 .WORD BIT9
 .WORD BIT10
 .WORD BIT11
 .WORD BIT12
 .WORD BIT13
 .WORD BIT14
 .WORD 77600 ;SHIFTING 0
 .WORD 77400
 .WORD 76600
 .WORD 75600
 .WORD 73600
 .WORD 67600
 .WORD 57600
 .WORD 37600
 .WORD 77600
 .WORD 200
 .WORD 600 ;GROWING 1
 .WORD 1600
 .WORD 3600
 .WORD 7600
 QUAMAX: .WORD 17600
 HALMAX: .WORD 37600
 .WORD 77600
 .WORD 77400 ;GROWING 0
 .WORD 77000
 .WORD 76000
 .WORD 74000
 .WORD 70000
 .WORD 60000
 .WORD 40000
 SKEND: .WORD 00000
 RL2: .WORD BIT15
 QMAX: .WORD 37600
 HMAX: .WORD 77600

.WORD 177600
 .WORD 177400
 .WORD 176600
 .WORD 173600
 .WORD 167600
 .WORD 157600
 .WORD 137600
 .WORD 177000
 .WORD 176000

255 002760 174000
 256 002762 170000
 257 002764 060000
 258 002766 040000
 259 002770 000000
 260
 261
 262
 263 002772 000000
 264 002774 000002
 265 002776 000004
 266 003000 000010
 267 003002 000020
 268 003004 000040
 269 003006 000100
 270 003010 000400
 271 003012 001000
 272 003014 001576
 273 003016 001574
 274 003020 001570
 275 003022 001560
 276 003024 001540
 277 003026 001500
 278 003030 001400
 279 003032 001576
 280 003034 001574
 281 003036 001566
 282 003040 001556
 283 003042 001536
 284 003044 001436
 285 003046 001136
 286 003050 000076
 287 003052 000006
 288 003054 000016
 289 003056 000036
 290 003060 000076
 291 003062 000176
 292 003064 000576
 293 003066 001576
 294 003070 000000
 295 003072 000000
 296 003074
 297 003274
 298 003774
 299

.WORD 174000
 .WORD 170000
 .WORD 60000
 .WORD 40000
 SKEEND: .WORD 000000

;PATTERNS FOR TEST OF RLCS

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 1536
 .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
 ERPOINT: .WORD 0
 ERCOUNT: .BLKW 64.
 HDRBUF: .BLKW 160.
 ENDMOD

	BGNMOD	GLBTXT			
	.SBTTL	GLOBAL	TEXT		
1	003774				
2					
3					
4					
5					
6					
7	003774	040	104	122	DEMES: .ASCIZ / DRV/
8	004001	040	116	130	NXMMES: .ASCIZ / NXM/
9	004006	040	117	120	OPIMES: .ASCIZ / OPI/
10	004013	040	110	103	HRCMES: .ASCIZ / HRCR/
11	004021	040	110	116	HNFMES: .ASCIZ / HNF/
12	004026	040	104	103	DCKMES: .ASCIZ / DCK/
13	004033	040	104	114	DLTMES: .ASCIZ / DLT/
14	004040	015	012	000	MSCRLF: .ASCIZ <15><12>
15	004043	015	000		LF: .ASCIZ <15>
16	004045	040	103	117	COMP: .ASCIZ / COMP/
17	004053	106	117	122	OPIERR: .ASCIZ /FORCED OPI(GET STATUS) CAUSED OTHER ERRORS/
18	004126	116	117	117	NOPMES: .ASCIZ /NOOP OPERATION-FLAG MODE/
19	004157	116	117	117	NOPINT: .ASCIZ /NOOP OPERATION-INTR. MODE/
20	004211	127	122	111	WCKMES: .ASCIZ /WRITE CHECK OPERATION-FLAG MODE/
21	004251	127	122	111	WCKINT: .ASCIZ /WRITE CHECK OPERATION-INTR. MODE/
22	004312	122	105	101	RHDMES: .ASCIZ /READ HEADER OPERATION-FLAG MODE/
23	004352	122	105	101	RHDINT: .ASCIZ /READ HEADER OPERATION-INTR. MODE/
24	004413	123	105	105	SEKMES: .ASCIZ /SEEK OPERATION-FLAG MODE/
25	004444	123	105	105	SEKINT: .ASCIZ /SEEK OPERATION-INTR. MODE/
26	004476	107	105	124	GSTMES: .ASCIZ /GET STATUS OPERATION-FLAG MODE/
27	004535	107	105	124	GSTINT: .ASCIZ /GET STATUS OPERATION-INTR MODE/
28	004574	103	123	072	ARLCS: .ASCIZ /CS: /
29	004601	040	102	101	ARLBA: .ASCIZ / BA: /
30	004607	040	104	101	ARLDA: .ASCIZ / DA: /
31	004615	040	115	120	ARLMP: .ASCIZ / MP: /
32	004623	102	105	106	BEREG: .ASCIZ /BEFORE COMMAND: /
33	004644	124	111	115	AFREG: .ASCIZ /TIME OF ERROR: /
34	004665	103	117	116	CRTIM: .ASCIZ /CONTROLLER TIMED OUT/
35	004712	104	122	111	DRTIM: .ASCIZ /DRIVE READY TIMED OUT/
36	004740	103	101	116	EM1: .ASCIZ /CAN NOT ADDRESS RLCS/
37	004765	103	101	116	EM2: .ASCIZ /CAN NOT ADDRESS RLBA/
38	005012	103	101	116	EM3: .ASCIZ /CAN NOT ADDRESS RLDA/
39	005037	103	101	116	EM4: .ASCIZ /CAN NOT ADDRESS RLMP/
40	005064	122	114	103	EM5: .ASCIZ %RLCS READ/WRITE ERROR (BIT 0 DON'T CARE)%
41	005135	122	114	102	EM6: .ASCIZ %RLBA READ/WRITE ERROR%
42	005163	122	114	104	EM7: .ASCIZ %RLDA READ/WRITE ERROR%
43	005211	117	120	111	EM11: .ASCIZ /OPI WOULD NOT GENERATE INTERRUPT/
44	005252	116	117	040	EM13: .ASCIZ /NO INTERRUPT FROM NOOP(0)/
45	005304	116	117	117	EM14: .ASCIZ /NOOP(0) MODIFIED RLMP/
46	005332	116	117	117	EM15: .ASCIZ /NOOP(0) MODIFIED RLBA/
47	005360	116	117	117	EM16: .ASCIZ /NOOP(0) MODIFIED RLDA/
48	005406	111	116	124	EM17: .ASCIZ /INTERRUPT PRIORITY FAILURE/
49	005441	107	105	124	EM30: .ASCIZ /GET STATUS WOULD NOT INTERRUPT/
50	005500	107	105	124	EM30A: .ASCIZ /GET STATUS SHOULD NOT INTERRUPT/
51	005540	122	114	115	EM32: .ASCIZ /RLMP CONTAINED WRONG STATUS/
52	005574	117	120	111	EM33: .ASCIZ /OPI DID NOT SET-GSTAT WITHOUT GS BIT/
53	005641	117	120	111	EM34: .ASCIZ /OPI DID NOT SET-GSTAT WITHOUT GS AND MK BITS/
54	005716	122	105	101	EM37: .ASCIZ /READ HEADER WOULD NOT INTERRUPT/
55	005756	102	101	104	EM41: .ASCIZ /BAD CYLINDER OR HEAD SELECT IN REPEATED READ HEADER TEST/
56	006047	102	101	104	EM42: .ASCIZ /BAD HEADER CRC ON READ HEADER/
57	006105	123	105	103	EM43: .ASCIZ /SECTOR ADDRESS OUT OF SEQUENCE DURING CONSECUTIVE READ HEADERS/
58	006204	127	122	111	EM44: .ASCIZ /WRITING RLMP MODIFIED RLCS/
59	006237	127	122	111	EM45: .ASCIZ /WRITING RLMP MODIFIED RLBA/
60	006272	127	122	111	EM46: .ASCIZ /WRITING RLMP MODIFIED RLDA/


```
61 006325      123      105      105  EM47:  .ASCIZ  /SEEK WOULD NOT INTERRUPT/
62 006356      104      122      111  EM52:  .ASCIZ  /DRIVE READY CAUSED EXTRANEIOUS INTERRUPT/
63 006426      102      101      104  EM54:  .ASCIZ  /BAD SEEK-TEST OF DIFFENCE WORD/
64 006465      102      101      104  EM55:  .ASCIZ  /BAD HEAD SELECT VIA RD HDR/
65 006520      102      101      104  EM56:  .ASCIZ  /BAD HEAD SELECT VIA GET STATUS/
66 006557      114      117      101  EM57:  .ASCII  /LOADING RLDA BEFORE DRIVE READY ON SEEK/<15><12>
67 006630      104      122      111  .ASCIZ  /DRIVE READY DID NOT SET/
68 006660      102      111      124  EM61:  .ASCIZ  /BIT SET INSTRUCTION ON RLCS YIELDED WRONG RESULT/
69 006741      102      111      124  EM62:  .ASCIZ  /BIT CLEAR INSTRUCTION ON RLCS YIELDED WRONG RESULT/
70 007024      102      111      124  EM63:  .ASCIZ  /BIT SET INSTRUCTION ON RLBA YIELDED WRONG RESULT/
71 007105      102      111      124  EM64:  .ASCIZ  /BIT CLEAR INSTRUCTION ON RLBA YIELDED WRONG RESULT/
72 007170      102      111      124  EM65:  .ASCIZ  /BIT SET INSTRUCTION ON RLDA YIELDED WRONG RESULT/
73 007251      102      111      124  EM66:  .ASCIZ  /BIT CLEAR INSTRUCTION ON RLDA YIELDED WRONG RESULT/
74 007334      102      125      123  EM67:  .ASCIZ  /BUS RESET DID NOT CLEAR RLCS/
75 007371      102      125      123  EM70:  .ASCIZ  /BUS RESET DID NOT CLEAR RLBA/
76 007426      102      125      123  EM71:  .ASCIZ  /BUS RESET DID NOT CLEAR RLDA/
77 007463      127      122      111  EM72:  .ASCIZ  /WRITING RLCS MODIFIED RLBA/
78 007516      127      122      111  EM73:  .ASCIZ  /WRITING RLCS MODIFIED RLDA/
79 007551      127      122      111  EM74:  .ASCIZ  /WRITING RLBA MODIFED RLCS/
80 007603      127      122      111  EM75:  .ASCIZ  /WRITING RLBA MODIFED RLDA/
81 007635      127      122      111  EM76:  .ASCIZ  /WRITING RLDA MODIFIED RLCS/
82 007670      127      122      111  EM77:  .ASCIZ  /WRITING RLDA MODIFIED RLBA/
83 007723      122      114      103  EM101: .ASCIZ  /RLCS CONTAINED FOLLOWING ERROR(S): /
84 007770      .BLKB   120.
85
86          .EVEN
87
91 010160
92          ENDMOD
```

1				.SBTTL	GLOBAL ERRORS		
2							
3	010160			BGNMOD	GLBERR		
4							
5	010160			BGNMSG	ERRO		
6							
7	010160	004737	010504	JSR	PC,LINE1		
8	010164	004737	010540	JSR	PC,LINE2		
9							
10	010170	004537	013034	JSR	R5,CKERLT		:CHECK ERROR LIMIT
11	010174			ENDMSG			
	010174			L10000:			
	010174	104423		TRAP	C\$MSG		
12							
13	010176			BGNMSG	ERR1		
14							
15	010176	004737	010504	JSR	PC,LINE1		
16							
17	010202	004537	013034	JSR	R5,CKERLT		:CHECK ERROR LIMIT
18	010206			ENDMSG			
	010206			L10001:			
	010206	104423		TRAP	C\$MSG		
19							
20	010210			BGNMSG	ERR2		
21							
22	010210	004737	010504	JSR	PC,LINE1		
23	010214			PRINTB	#FRMT4,GDDAT,BDDAT		
	010214	013746	002364	MOV	BDDAT,-(SP)		
	010220	013746	002362	MOV	GDDAT,-(SP)		
	010224	012746	011137	MOV	#FRMT4,-(SP)		
	010230	012746	000003	MOV	#3,-(SP)		
	010234	010600		MOV	SP,R0		
	010236	104414		TRAP	C\$PNTB		
	010240	062706	000010	ADD	#10,SP		
24							
25	010244	004537	013034	JSR	R5,CKERLT		:CHECK ERROR LIMIT
26	010250			ENDMSG			
	010250			L10002:			
	010250	104423		TRAP	C\$MSG		
27							
28	010252			BGNMSG	ERR3		
29							
30	010252	004737	010504	JSR	PC,LINE1		
31	010256	004737	010540	JSR	PC,LINE2		
32	010262			PRINTB	#FRMT5,TMPO,BDDAT,GDDAT		
	010262	013746	002362	MOV	GDDAT,-(SP)		
	010266	013746	002364	MOV	BDDAT,-(SP)		
	010272	013746	002354	MOV	TMPO,-(SP)		
	010276	012746	011175	MOV	#FRMT5,-(SP)		
	010302	012746	000004	MOV	#4,-(SP)		
	010306	010600		MOV	SP,R0		
	010310	104414		TRAP	C\$PNTB		
	010312	062706	000012	ADD	#12,SP		
33							
34	010316	004537	013034	JSR	R5,CKERLT		:CHECK ERROR LIMIT
35	010322			ENDMSG			
	010322			L10003:			

36	010322	104423		TRAP	C\$MSG	
37	010324			BGNMSG	ERR4	
38						
39	010324	004737	010504	JSR	PC,LINE1	
40	010330	004737	010540	JSR	PC,LINE2	
41	010334			PRINTB	#FRMT4,GDDAT,BDDAT	
	010334	013746	002364	MOV	BDDAT,-(SP)	
	010340	013746	002362	MOV	GDDAT,-(SP)	
	010344	012746	011137	MOV	#FRMT4,-(SP)	
	010350	012746	000003	MOV	#3,-(SP)	
	010354	010600		MOV	SP,R0	
	010356	104414		TRAP	C\$PNTB	
	010360	062706	000010	ADD	#10,SP	
42						
43	010364	004537	013034	JSR	R5,CKERLT	;CHECK ERROR LIMIT
44	010370			ENDMSG		
	010370			L10004:		
	010370	104423		TRAP	C\$MSG	
45						
46	010372			BGNMSG	ERR5	
47						
48	010372	004737	010504	JSR	PC,LINE1	
49						
50	010376	004537	013034	JSR	R5,CKERLT	;CHECK ERROR LIMIT
51	010402			ENDMSG		
	010402			L10005:		
	010402	104423		TRAP	C\$MSG	
52						
53	010404			BGNMSG	ERR6	
54						
55	010404	004737	010504	JSR	PC,LINE1	
56	010410	004737	010752	JSR	PC,LINE3	
57	010414	004737	010540	JSR	PC,LINE2	
58						
59						
60	010420			1\$:	PRINTB	#FRMT99
	010420	012746	011172	MOV	#FRMT99,-(SP)	
	010424	012746	000001	MOV	#1,-(SP)	
	010430	010600		MOV	SP,R0	
	010432	104414		TRAP	C\$PNTB	
	010434	062706	000004	ADD	#4,SP	
61	010440	004537	013034	JSR	R5,CKERLT	;CHECK ERROR LIMIT
62	010444			ENDMSG		
	010444			L10006:		
	010444	104423		TRAP	C\$MSG	
63						
64	010446			BGNMSG	ERR7	
65						
66	010446	004737	010504	JSR	PC,LINE1	
67	010452			PRINTB	#FRMT6,BDDAT	
	010452	013746	002364	MOV	BDDAT,-(SP)	
	010456	012746	011246	MOV	#FRMT6,-(SP)	
	010462	012746	000002	MOV	#2,-(SP)	
	010466	010600		MOV	SP,R0	
	010470	104414		TRAP	C\$PNTB	
	010472	062706	000006	ADD	#6,SP	

68					
69	010476	004537	013034	JSR	R5,CKERLT
70					
71	010502			ENDMSG	
	010502			L10007:	
	010502	104423		TRAP	C\$MSG
72					
73	010504			LINE1:	PRINTB #FRMT1,RLCS,<B,DRIVE+1>
	010504	005046		CLR	-(SP)
	010506	153716	002271	BISB	DRIVE+1,(SP)
	010512	013746	002250	MOV	RLCS,-(SP)
	010516	012746	011024	MOV	#FRMT1,-(SP)
	010522	012746	000003	MOV	#3,-(SP)
	010526	010600		MOV	SP,R0
	010530	104414		TRAP	C\$PNTB
	010532	062706	000010	ADD	#10,SP
74	010536	000207		RTS	PC
75					
76	010540			LINE2:	PRINTB #FRMT2,#BEREG,#ARLCS,B.CS,#ARLBA,B.BA
	010540	013746	002274	MOV	B.BA,-(SP)
	010544	012746	004601	MOV	#ARLBA,-(SP)
	010550	013746	002272	MOV	B.CS,-(SP)
	010554	012746	004574	MOV	#ARLCS,-(SP)
	010560	012746	004623	MOV	#BEREG,-(SP)
	010564	012746	011064	MOV	#FRMT2,-(SP)
	010570	012746	000006	MOV	#6,-(SP)
	010574	010600		MOV	SP,R0
	010576	104414		TRAP	C\$PNTB
	010600	062706	000016	ADD	#16,SP
77	010604			PRINTB	#FRMT2A,#ARLDA,B.DA,#ARLMP,B.MP
	010604	013746	002300	MOV	B.MP,-(SP)
	010610	012746	004615	MOV	#ARLMP,-(SP)
	010614	013746	002276	MOV	B.DA,-(SP)
	010620	012746	004607	MOV	#ARLDA,-(SP)
	010624	012746	011103	MOV	#FRMT2A,-(SP)
	010630	012746	000005	MOV	#5,-(SP)
	010634	010600		MOV	SP,R0
	010636	104414		TRAP	C\$PNTB
	010640	062706	000014	ADD	#14,SP
78	010644			PRINTB	#FRMT2,#AFREG,#ARLCS,E.CS,#ARLBA,E.BA
	010644	013746	002310	MOV	E.BA,-(SP)
	010650	012746	004601	MOV	#ARLBA,-(SP)
	010654	013746	002306	MOV	E.CS,-(SP)
	010660	012746	004574	MOV	#ARLCS,-(SP)
	010664	012746	004644	MOV	#AFREG,-(SP)
	010670	012746	011064	MOV	#FRMT2,-(SP)
	010674	012746	000006	MOV	#6,-(SP)
	010700	010600		MOV	SP,R0
	010702	104414		TRAP	C\$PNTB
	010704	062706	000016	ADD	#16,SP
79	010710			PRINTB	#FRMT2B,#ARLDA,E.DA,#ARLMP,E.MP
	010710	013746	002314	MOV	E.MP,-(SP)
	010714	012746	004615	MOV	#ARLMP,-(SP)
	010720	013746	002312	MOV	E.DA,-(SP)
	010724	012746	004607	MOV	#ARLDA,-(SP)
	010730	012746	011116	MOV	#FRMT2B,-(SP)
	010734	012746	000005	MOV	#5,-(SP)

80	010740	C10600			MOV	SP,R0	
	010742	104414			TRAP	C\$PNTB	
	010744	062706	000014		ADD	#14,SP	
	010750	000207			RTS	PC	
81							
82	010752			LINE3:	PRINTB	#FRMT3,#EM101	
	010752	012746	007723		MOV	#EM101,-(SP)	
	010756	012746	011132		MOV	#FRMT3,-(SP)	
	010762	012746	000002		MOV	#2,-(SP)	
	010766	010600			MOV	SP,R0	
	010770	104414			TRAP	C\$PNTB	
	010772	062706	000006		ADD	#6,SP	
83	010776				PRINTB	#FRMT3,#EM102	
	010776	012746	007770		MOV	#EM102,-(SP)	
	011002	012746	011132		MOV	#FRMT3,-(SP)	
	011006	012746	000002		MOV	#2,-(SP)	
	011012	010600			MOV	SP,R0	
	011014	104414			TRAP	C\$PNTB	
	011016	062706	000006		ADD	#6,SP	
84	011022	000207			RTS	PC	
85							
89							
90	011024	045	101	103	FRMT1:	.ASCIZ	/%ACONTROLLER: %06%A DRIVE: %01/
91	011064	045	116	045	FRMT2:	.ASCIZ	/%N%T%T%06%T%06/
92	011103	045	124	045	FRMT2A:	.ASCIZ	/%T%06%T%06/
93	011116	045	124	045	FRMT2B:	.ASCIZ	/%T%06%T%06%T%06%T%06/
94	011132	045	116	045	FRMT3:	.ASCIZ	/%N%T/
95	011137	045	116	045	FRMT4:	.ASCIZ	/%N%AEXP'D: %06%A REC'D: %06/
96	011172	045	116	000	FRMT99:	.ASCIZ	/%N/
97	011175	045	116	045	FRMT5:	.ASCIZ	/%N%ALAST: %06%A PRES: %06%A EXP'D: %06%N/
98	011246	045	116	045	FRMT6:	.ASCIZ	/%N%AAT PROCESSOR LEVEL %06%N/
99	011303	045	101	105	FRMT11:	.ASCIZ	/%AERROR LIMIT EXCEEDED-DROPPED%N/
100	011344	045	116	045	FRMT12:	.ASCIZ	/%N%ADRIVE DID NOT RECOVER FROM POWER FAILURE%N/
101	011423	045	116	045	FRMT13:	.ASCIZ	/%N%T%A - WILL NOT TEST%N/
102	011454	045	116	045	FRMT14:	.ASCIZ	/%N%ADRIVE DROPPED - NO CONTROLLER%N/
103	011520	045	116	045	FRMT15:	.ASCIZ	/%N%ADRIVE DROPPED - DID NOT RESPOND WITH 'READY'%N/
104							
105					.EVEN		
106							
107							
111							
112							
113							
114	011604				ENDMOD		
115							
116	011604				BGNMOD	HPTCODE	
117							
118	011604				BGNHW		:DEFAULT HARDWARE TABLE
	011604	000006			.WORD	L10010-LSHW/2	
119	011606	174400			.WORD	174400	:CSR
120	011610	000160			.WORD	160	:VECTOR
121	011612	000240			.WORD	240	:PRIORITY
122	011614	000001			.WORD	1	:RL01 = 1
123	011616	000000			.WORD	0	:DRIVE (BITS 8,9,10)
124	011620	000001			.WORD	1	:RL11 = 1, RLV11 = 0
125							
126	011622				ENDHW		

127	011622		L10010:	
128	011622		ENDMOD	
129				
130	011622		BGNMOD SPTCODE	
131				
132	011622		BGNSW	:DEFAULT SOFTWARE TABLE
	011622	000003	.WORD	L10011-LSSW/2
133				
134	011624	000000	DROP: .WORD	0
135	011626	000012	MERLMT: .WORD	10.
136	011630	000000	T.SIZE: .WORD	0
137				
138	011632		ENDSW	
	011632		L10011:	
139				
140	011632		ENDMOD	
141				
142	011632		BGNMOD DSPCODE	
143				
144	011632		DISPATCH	44
	011632	000054	.WORD	44
	011634	014406	.WORD	T1
	011636	014502	.WORD	T2
	011640	014576	.WORD	T3
	011642	014672	.WORD	T4
	011644	014766	.WORD	T5
	011646	015106	.WORD	T6
	011650	015212	.WORD	T7
	011652	015300	.WORD	T8
	011654	015424	.WORD	T9
	011656	015550	.WORD	T10
	011660	015656	.WORD	T11
	011662	015756	.WORD	T12
	011664	016046	.WORD	T13
	011666	016146	.WORD	T14
	011670	016256	.WORD	T15
	011672	016332	.WORD	T16
	011674	016370	.WORD	T17
	011676	016514	.WORD	T18
	011700	016654	.WORD	T19
	011702	017014	.WORD	T20
	011704	017220	.WORD	T21
	011706	017252	.WORD	T22
	011710	017460	.WORD	T23
	011712	017546	.WORD	T24
	011714	017714	.WORD	T25
	011716	017744	.WORD	T26
	011720	020116	.WORD	T27
	011722	020204	.WORD	T28
	011724	020332	.WORD	T29
	011726	020354	.WORD	T30
	011730	020434	.WORD	T31
	011732	020600	.WORD	T32
	011734	020736	.WORD	T33
	011736	021254	.WORD	T34
	011740	021350	.WORD	T35

011742	021414	.WORD	T36
011744	021540	.WORD	T37
011746	022156	.WORD	T38
011750	022310	.WORD	T39
011752	022452	.WORD	T40
011754	022612	.WORD	T41
011756	022764	.WORD	T42
011760	023412	.WORD	T43
011762	024132	.WORD	T44
145			
146	011764	ENDMOD	
147			
148			

```

1
2
3
4 011764 000000 .SBTTL LOAD PROTECTION TABLE
5 011766 177777 BGNPROT
6 011770 000012 .WORD 0 ;P-TABLE OFFSET OF CSR
7 011772 .WORD -1 ;NOT A MASS-BUS DRIVE
8
9 .SBTTL INITIALIZATION CODE
10 011772 BGNMOD INITCODE
11
12 011772 BGNINIT
13
14 011772 BRESET
15 011774 104433 TRAP CSRESET
16 012000 104447 000034 READEF #EF.PWR ;POWER UP?????
17 012002 103004 002012 002242 MOV #EF.PWR,RO
18 012012 000475 TRAP CSREFG ;NO,BRANCH
19 012014 012700 000037 NOPWR: BNCOMPLETE NOPWR ;YES, SET POWER FLAG
20 012022 103404 BCC NOPWR ;GO TO CONTINUE POINT
21 012024 012700 000040 READEF #EF.RESTART ;RESTART?
22 012030 104447 MOV #EF.RESTART,RO
23 012032 103010 TRAP CSREFG
24 012034 012700 003074 BNCOMPLETE START1
25 012040 012701 000100 BCC CONTINUE
26 012044 005020 START1: MOV #ERCOUNT,RO
27 012046 005301 1$: CLR (RO)+
28 012050 001375 DEC R1
29 012052 000407 BNE 1$
30 012054 012700 000036 CONTINUE: READEF #EF.CONTINUE ;CONTINUE????
31 012060 104447 MOV #EF.CONTINUE,RO
32 012062 103451 TRAP CSREFG
33 012064 005737 002244 BNCOMPLETE CONT
34 012070 001011 BCS CONT
35 012072 012737 177777 002246 NXT: TST UUT ;DONE ALL UUT'S
36 012100 013737 002012 002244 BNE XXX ;NO
37 012106 012737 003072 003072 START: MOV #-1,UNITST
38
39 012114 005237 002246 XXX: INC UNITST
40 012120 062737 000002 003072 ADD #2,ERPOINT
41 012126 005337 002244 DEC UUT
42 012132 013700 002246 REST: GPHARD UNITST,RO
    012132 104442 MOV UNITST,RO
    TRAP CS$GPHRD
    
```



```

43 012140          BCOMPLETE      1$
    012140 103406          BCS      1$
44 012142 005737 002242      TST      PWRFLG      ;POWER FLAG TO 0
45 012146 001746          BEQ      NXT          ;YES, DONT DEC IT
46 012150 005337 002242      DEC      PWRFLG
47 012154 000743          BR      NXT          ;GET NEXT ONE
48
49 012156 012037 002262      1$:    MOV      (R0)+,BCSR
50 012162 012037 002266      MOV      (R0)+,BVEC
51 012166 012037 002264      MOV      (R0)+,BPRIOR
52 012172 012037 002406      MOV      (R0)+,T.DRIVE
53 012176 012037 002270      MOV      (R0)+,DRIVE
54 012202 012037 002410      MOV      (R0)+,T.CNTRLR      ;GET CONTROLLER TYPE
55
56 012206 013700 002262      CONT:  MOV      BCSR,R0      ;BUILD LOGICAL ADDRESSES OF REGISTERS
57 012212 010037 002250      MOV      R0,RLCS
58 012216 062700 000002      ADD      #2,R0
59 012222 010037 002252      MOV      R0,RLBA
60 012226 062700 000002      ADD      #2,R0
61 012232 010037 002254      MOV      R0,RLDA
62 012236 062700 000002      ADD      #2,R0
63 012242 010037 002256      MOV      R0,RLMP
64 012246 022737 000002 002410  CMP      #2,T.CNTRLR      ;IF THIS IS AN RLV12, BUILD LOGICAL
65 012254 001004          BNE      1$          ;ADDRESS FOR BUS ADDRESS EXTENSION.
66 012256 062700 000002      ADD      #2,R0
67 012262 010037 002260      MOV      R0,RLBE
68
69 012266 005737 002242      1$:    TST      PWRFLG      ;RECENT POWER FAILURE?
70 012272 001476          BEQ      END          ;NO
71
72      ;THERE WAS A RECENT POWER FAILURE, THEREFORE WE WILL WAIT
73      ;FOR THE DRIVE TO COME READY
74
75 012274 012701 000170          MOV      #120.,R1      ;INITIALIZE WAIT COUNT
76 012300 012777 000200 167742  MOV      #200,@RLCS      ;SET CRDY
77 012306 053777 002270 167734  BIS      DRIVE,@RLCS      ;SET IN DRIVE SELECT
78 012314 032777 000001 167726  DRVRDY: BIT      #DRDY,@RLCS      ;DRIVE READY???
79 012322 001042          BNE      BGNTST      ;YES, THEN START TEST
80 012324 012737 000050 002414  MOV      #40.,DLYCNT      ;INITIALIZE DELAY COUNT
81 012332          WAITO:  DELAY      1      ;IMPLEMENT 100-USEC DELAY
    012332 012727 000001      MOV      #1,(PC)+
    012336 000000          .WORD      0
    012340 013727 002116      MOV      LSDLY,(PC)+
    012344 000000          .WORD      0
    012346 005367 177772      DEC      -6(PC)
    012352 001375          BNE      -.4
    012354 005367 177756      DEC      -22(PC)
    012360 001367          BNE      -.20
82 012362 005337 002414      DEC      DLYCNT      ;DECREMENT DELAY COUNT
83 012366 001361          BNE      WAITO      ;BRANCH IF TIME DELAY NOT EXPIRED
84 012370 005301          DEC      R1          ;SIXTY SECONDS GONE BY
85 012372 001350          BNE      DRVRDY      ;NO, GO BACK
86 012374          PRINTB #FRMT12      ;DROPPING DRIVE - DRIVE DID NOT RECOVER
    012374 012746 011344      MOV      #FRMT12,-(SP)
    012400 012746 000001      MOV      #1,-(SP)
    012404 010600          MOV      SP,R0
    012406 104414          TRAP     C$PNTB
    
```

```

87 012410 062706 000004          ADD      #4,SP          ;/FROM POWER FAILURE
88 012414 004737 010504          6$:      JSR      PC,LINE1      ;GIVE DRIVE INFO
89 012420 013700 002246          DODU     UNITST          ;TELL SUPERVISOR TO DROP IT
      012424 104451          MOV      UNITST,RO
90 012426 104444          TRAP    CSDODU
      012426 104444          DOCLN   CSDCLN          ;FORCE AN ABORT
91 012430 012777 000013 167616 BGNTST:  MOV      #13,@RLDA          ;SETUP DR RST
92 012436 012777 000204 167604      MOV      #204,@RLCS        ;GS FUNC
93 012444 053777 002270 167576      BIS      DRIVE,@RLCS        ;SELECT DRIVE
94 012452 042777 000200 167570      BIC      #200,@RLCS        ;ISSUE IT
95 012460 032777 000200 167562      4$:      BIT      #200,@RLCS        ;WAIT FOR READY
96 012466 001774          BEQ     4$
97 012470          END:     SETVEC   BVEC,#INTSRV,#340
      012470 012746 000340          MOV      #340,-(SP)
      012474 012746 014212          MOV      #INTSRV,-(SP)
      012500 013746 002266          MOV      BVEC,-(SP)
      012504 012746 000003          MOV      #3,-(SP)
      012510 104437          TRAP    C$SVEC
      012512 062706 000010          ADD      #10,SP
98 012516 005037 002324          CLR     PFLG          ;CLR PROCESSOR FLAG
99 012522          READBUS          ;Q-BUS
      012522 104407          TRAP    C$RDBU
100 012524          BNCOMPLETE 1$
      012524 103002          BCC     1$
101 012526 005237 002324          INC     PFLG          ;NO, Q-BUS THEN
102 012532          1$:
103 012532          ENDINIT
      012532 104411          L10013: TRAP    C$INIT
104          ENDMOD
105 012534
106
107          .SBTTL AUTO DROP SECTION
108 012534          BGNAUTO
109 012534 005037 002326          CLR     TRPFLG          ;CLEAR TRAP FLAG
110          ;SET UP VECTOR TO DETECT NON-EXISTENT
111          ;/CONTROLLER
112 012540          SETVEC   ERRVEC,#TRPHAN,#340
      012540 012746 000340          MOV      #340,-(SP)
      012544 012746 014204          MOV      #TRPHAN,-(SP)
      012550 013746 002340          MOV      ERRVEC,-(SP)
      012554 012746 000003          MOV      #3,-(SP)
      012560 104437          TRAP    C$SVEC
      012562 062706 000010          ADD      #10,SP
113 012566 012746 000340          MOV      #340,-(SP)
114 012572 012746 014204          MOV      #TRPHAN,-(SP)
115 012576 013746 002340          MOV      ERRVEC,-(SP)
116 012602 012746 000003          MOV      #3,-(SP)
117 012606 104037          EMT     C$SVEC
118 012610 062706 000010          ADD      #10,SP
119
120 012614 005777 167430          TST     @RLCS          ;ACCESS CONTROLLER
121 012620          CLRVEC   ERRVEC          ;RELEASE VECTOR
      012620 013700 002340          MOV      ERRVEC,RO
      012624 104436          TRAP    C$CVEC
    
```


122	012626	013700	002340		MOV	ERRVEC,RO		
123	012632	104036			EMT	C\$CVEC		
124	012634	005737	002326		TST	TRPFLG	:DID IT TRAP?	
125	012640	001416			BEQ	1\$:NO - CHECK ITS DRIVE	
126	012642				PRINTB	#FRMT14	:ELSE, PRINT MSG. "DRIVE DROPPED - NO CONTROLLER"	
	012642	012746	011454		MOV	#FRMT14, -(SP)		
	012646	012746	000001		MOV	#1, -(SP)		
	012652	010600			MOV	SP,RO		
	012654	104414			TRAP	C\$PNTB		
	012656	062706	000004		ADD	#4,SP		
127	012662	004737	010504		JSR	PC,LINE1	:PROVIDE DRIVE INFORMATION	
128	012666				DODU	UNITST	:DO DROP UNIT ON DRIVE	
	012666	013700	002246		MOV	UNITST,RO		
	012672	104451			TRAP	C\$DODU		
129	012674	000427			BR	2\$:EXIT	
130								
131	012676	012777	000200	167344	1\$:	MOV	#200,@RLCS	:SET CONTROLLER READY
132	012704	053777	002270	167336		BIS	DRIVE,@RLCS	:SELECT DRIVE
133	012712	032777	000001	167330		BIT	#1,@RLCS	:IS DRIVE READY?
134	012720	001015				BNE	2\$:YES - EXIT
135								:ELSE, PRINT MSG. "DRIVE DROPPED - DID NOT
136								:RESPOND WITH "READY"
137	012722				PRINTB	#FRMT15		
	012722	012746	011520		MOV	#FRMT15, -(SP)		
	012726	012746	000001		MOV	#1, -(SP)		
	012732	010600			MOV	SP,RO		
	012734	104414			TRAP	C\$PNTB		
	012736	062706	000004		ADD	#4,SP		
138	012742	004737	010504		JSR	PC,LINE1	:PROVIDE DRIVE INFORMATION	
139	012746				DODU	UNITST	:DO DROP UNIT ON DRIVE	
	012746	013700	002246		MOV	UNITST,RO		
	012752	104451			TRAP	C\$DODU		
140	012754				2\$:			
141	012754				ENDAUTO			
	012754				L10014:			
	012754	104461			TRAP	C\$AUTO		
142								
143	012756				BGNMOD	CLNCODE		
144								
145	012756				BGNCLN			
146								
147	012756				SETPRI	#PRI07		
	012756	012700	000340		MOV	#PRI07,RO		
	012762	104441			TRAP	C\$SPRI		
148								
149	012764	032777	000200	167256	1\$:	BIT	#CRDY,@RLCS	
150	012772	001774				BEQ	1\$	
151								
152	012774	042777	000100	167246		BIC	#INTEN,@RLCS	
153								
154	013002				CLRVEC	BVEC		
	013002	013700	002266		MOV	BVEC,RO		
	013006	104436			TRAP	C\$CVEC		
155								
156								
157								
158	013010	005737	002242		TST	PWRFLG	:TREAT POWER FAILURE	

159	013014	001402	BEQ	2\$
160				
161	013016	005337 002242	DEC	PWRFLG
162				
163	013022		2\$:	ENDCLN
164	013022		L10015:	TRAP C\$CLEAN
	013022	104412		
165				
166	013024		ENDMOD	
167				
168				
169				
170	013024		BGNMOD	DRPCODE
171				
172	013024		BGNDU	
173				
174	013024	000240	NOP	
175				
176	013026		L10016:	ENDDU
	013026			
	013026	104453	TRAP	C\$DU
177				
178	013030		ENDMOD	
179				
180	013030		BGNMOD	ADDCODE
181				
182	013030		BGNAU	
183				
184	013030	000240	NOP	
185				
186	013032		L10017:	ENDAU
	013032			
	013032	104452	TRAP	C\$AU
187				
188	013034		ENDMOD	
189				
190				
191				


```

1
2
3
4 013034
5
6
7 013034
8 013034 104420
9 013036 103427
10 013040 005737 011624
11 013044 001424
12 013046 005277 170020
13 013052 027737 170014 011626
14 013060 002416
15 013062
16 013062 012746 011303
17 013066 012746 000001
18 013072 010600
19 013074 104417
20 013076 062706 000004
21 013102 004737 010504
22 013106
23 013106 013700 002246
24 013112 104451
25 013114
26 013114 104444
27 013116
28 013116 000205
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44 013120 005037 002304
45 013124 032737 176000 002306
46 013132 001001
47 013134 000205
    
```

.SBTTL GLOBAL SUBROUTINES

BGNMOD GLBSUB

CKERLT: INLOOP
 TRAP C\$INLP
 BCOMPLETE 99\$
 BCS 99\$
 TST DROP
 BEQ 99\$
 INC @ERPOINT
 CMP @ERPOINT,MERLMT
 BLT 99\$

PRINTF #FRMT11
 MOV #FRMT11,-(SP)
 MOV #1,-(SP)
 MOV SP,R0
 TRAP C\$PNTF
 ADD #4,SP
 JSR PC,LINE1 ;DROP THE UNIT
 DODU UNITST
 MOV UNITST,R0
 TRAP C\$DODU
 DOCLN
 TRAP C\$DCLN

99\$: RTS R5

.SBTTL ROUTINE TO CHECK FOR CONTROLLER ERRORS

 *THIS ROUTINE WILL CHECK RLCS FOR ERRORS AND PRINT THEM
 *ACCORDINGLY. IT WILL MERGE THE ERROR PRINTOUT WITH THE TEST
 *ERROR MESSAGE.

*EXAMPLE: RLCS CONTAINED FOLLOWING ERROR(S):
 * DRV OPI HCRC HNF
 * SEEK UNDER INTERRUPT

*ROUTINE USES R0,R1 AND PICKS HEADER FROM R3

* CALL JSR R5,CHERR

CHERR: CLR DERFLG ;CLEAR OUT DRIVE ERROR FLAG
 BIT #176000,E.CS ;ANY ERRORS SET
 BNE 99\$;IF YES, INVESTIGATE
 RTS R5 ;NO, EXIT

```

48 013136 023727 002412 000004 199$: CMP      TMPFNC,#GSTAT      ;FUNCTION-NOP, RESET, GETSTATUS
49 013144 002401          BLT      98$          ;YES, GO CHECK IF ONLY DRIVE ERROR
50 013146 000414          BR       1$          ;YES SERVICE ERROR
51 013150 023727 002412 000002 98$:  CMP      TMPFNC,#WRCHK
52 013156 001410          BEQ     1$
53 013160 013700 002306      MOV     E.CS,R0      ;GET E.CS
54 013164 042700 001777      BIC     #1777,R0
55 013170 022700 140000      CMP     #140000,R0
56 013174 001001          BNE    1$
57 013176 000205          RTS     R5          ;NO, GO SERVICE
                    ;YES, EXIT
58
59 013200 012701 007770      1$:  MOV     #EM102,R1    ;GET START OF STRING
60 013204 005737 002306      TST     E.CS        ;IS COMPOSITE ERROR SET?(BETTER BE)
61 013210 100003          BPL     99$        ;IT'S NOT SOMETHING IS WRONG
62 013212 004537 013664      JSR     R5,FIX      ;YES, PUT "COMP" IN STRING
63 013216 004045          COMP
64 013220 032737 040000 002306 99$:  BIT     #DERR,E.CS  ;DRIVE ERROR SET?
65 013226 001405          BEQ     3$        ;NO, CONTINUE
66 013230 005237 002304      INC     DERFLG      ;SET DRV ERROR FLAG
67 013234 004537 013664      JSR     R5,FIX      ;YES, PUT "DRV" INTO STRING
68 013240 003774          DEMES
69 013242 032737 020000 002306 3$:  BIT     #NXM,E.CS  ;NON-EXISTENT MEMORY ERROR?
70 013250 001403          BEQ     4$        ;NO, CONTINUE
71 013252 004537 013664      JSR     R5,FIX      ;YES, PUT "NXM" INTO STRING
72 013256 004001          NXMMES
73 013260 032737 002000 002306 4$:  BIT     #OPI,E.CS  ;IS OPI SET?
74 013266 001422          BEQ     6$        ;NO, GO CHECK BITS 11 & 12
75 013270 004537 013664      JSR     R5,FIX      ;PUT "OPI" INTO STRING
76 013274 004006          OPIMES
77 013276 032737 004000 002306      BIT     #BIT11,E.CS ;HEADERCRC ERROR?
78 013304 001403          BEQ     5$        ;NO, GO CHECK HEADER NOT FOUND
79 013306 004537 013664      JSR     R5,FIX      ;GO PUT "HCRC" IN STRING
80 013312 004013          HCRCMES
81 013314 032737 010000 002306 5$:  BIT     #BIT12,E.CS ;HEADER NOT FOUND?
82 013322 001422          BEQ     8$        ;NO, GO PUT "CRLF" IN STRING
83 013324 004537 013664      JSR     R5,FIX      ;PUT "HNF" IN STRING
84 013330 004021          HNFMES
85 013332 000416          BR      8$        ;PUT "CRLF" IN STRING
86 013334 032737 004000 002306 6$:  BIT     #BIT11,E.CS ;DATA CRC ERROR?
87 013342 001403          BEQ     7$        ;NO, GO CHECK DATA LATE
88 013344 004537 013664      JSR     R5,FIX      ;PUT "DCK" IN STRING
89 013350 004026          DCKMES
90 013352 032737 010000 002306 7$:  BIT     #BIT12,E.CS ;DATA LATE ERROR?
91 013360 001403          BEQ     8$        ;NO, GO PUT IN "CRLF"
92 013362 004537 013664      JSR     R5,FIX      ;PUT "DLT" IN STRING
93 013366 004033          DLTMES
94 013370 004537 013664          8$:  JSR     R5,FIX
95 013374 004040          MSCRLF
96 013376 004537 013664      JSR     R5,FIX
97 013402 000000          RESTMS: .WORD 0      ;HEADER FROM TEST
98 013404 105011          CLRB   (R1)      ;PUT TERMINATOR IN
99
100 013406          ERRDF 300,LF,ERR6
    013406 104455      TRAP  CSERDF
    013410 000454      .WORD 300
    013412 004043      .WORD LF
    013414 010404      .WORD ERR6
    
```



```

101
102 013416 000205          RTS      R5          ;EXIT ROUTINE
103
104          .SBTTL  LOAD RLCS
105          *****
106          * ROUTINE TO LOAD RLCS WITH FUNCTION TO BE PERFORMED
107          * CALL:   JSR      R5,LDFUNC
108          *          .WORD          ;BITS TO BE LOADED, FUNCTION
109          *          ;AND INTR ENABLE ONLY
110          *
111          *
112          *
113 013420 012537 002332    LDFUNC: MOV      (R5)+,LDCSR    ;GET BITS TO LOAD
114 013424 005737 002304          TST      DERFLG
115 013430 001424          BEQ      98$
116 013432 013746 002272    MOV      B.CS,-(SP)
117 013436 012777 000013 166610    MOV      #13,@RLDA
118 013444 012737 000004 002272    MOV      #GSTAT,B.CS
119 013452 053737 002270 002272    BIS      DRIVE,B.CS
120 013460 013777 002272 166562    MOV      B.CS,@RLCS
121 013466 012637 002272          MOV      (SP)+,B.CS
122 013472 032777 000200 166550 99$:   BIT      #200,@RLCS
123 013500 001774          BEQ      99$
124 013502 010346          MOV      R3,-(SP)          ;SAVE R3
125 013504 042737 177661 002332 98$:   BIC      #177661,LDCSR    ;CLEAR ALL BUT FUNC & INTR EN
126 013512 013737 002332 013636    MOV      LDCSR,FNDFNC    ;SAVE FUNCTION
127 013520 042737 000100 013636    BIC      #INTEN,FNDFNC   ;ONLY FUNCTION
128 013526 013737 013636 002412    MOV      FNDFNC,TMPFNC
129 013534 012703 013640          MOV      #HDRLST,R3      ;GET HEADER LIST
130 013540 006237 013636          ASR      FNDFNC          ;ALIGN TO RIGHT
131 013544 001404          BEQ      2$
132 013546 022323          CMP      (R3)+,(R3)+     ;BUMP R3 BY 4
133 013550 005337 013636          DEC      FNDFNC          ;FOUND IT
134 013554 001374          BNE      1$              ;NO,KEEP LOOKING
135 013556 032737 000100 002332 2$:   BIT      #INTEN,LDCSR    ;YES,DO WE WANT FLAG OR INTR
136 013564 001401          BEQ      3$              ;FLAG BRANCH
137 013566 005723          TST      (R3)+          ;INTR POINT TO THAT ONE
138 013570 011303          MOV      (R3),R3        ;SET HEADER
139 013572 010337 013402          MOV      R3,RESTMS      ;SET UP HEADER
140 013576 053737 002270 002332    BIS      DRIVE,LDCSR    ;SELECT DRIVE
141 013604 052737 000200 002332 4$:   BIS      #200,LDCSR     ;CONTROLLER READY
142 013612 013777 002332 166430    MOV      LDCSR,@RLCS
143 013620 004537 013676          JSR      R5,BEFORE
144 013624 042777 000200 166416 5$:   BIC      #200,@RLCS
145 013632 012603          MOV      (SP)+,R3      ;RESTORE R3
146 013634 000205          RTS      R5            ;EXIT
147
148 013636 000000          FNDFNC: .WORD 0
149
150 013640 004126          HDRLST: NOPMES
151 013642 004157          NOPINT
152 013644 004211          WCKMES
153 013646 004251          WCKINT
154 013650 004476          OKHDR:  GSTMES
155 013652 004535          GSTINT
156 013654 004413          SEKMES
157 013656 004444          SEKINT
  
```

```

158 013660 004312          RHDMES
159 013662 004352          RHDINT
160
161          :*****
162          :*ROUTINE TO MOVE ASCII STRINGS
163          :*USES REGISTERS R1 - WHERE STRING IS BEING BUILT
164          :*
165          :*      CALL      JSR      R5, FIX
166          :*      .WORD      .WORD      ;ADDRESS OF STRING TO MOVE
167
168 013664 012500          FIX:  MOV      (R5)+, R0      ;GET ADDRESS AND MOVE RETURN
169 013666 112021          1$:  MOVB     (R0)+, (R1)+    ;GET BYTE AND UPDATE
170 013670 001376          BNE     1$                ;WATCH 0 BYTE TERMINATOR
171 013672 105741          TSTB    -(R1)             ;BACK UP OVER ZERO BYTE
172 013674 000205          RTS     R5                ;EXIT
173
174
175          ;LOAD REGISTERS BEFORE OPERATION
176          ;CALL:  JSR      R5,BEFORE
177
178 013676 017737 166346 002272  BEFORE:  MOV      @RLCS,B.CS      ;READ CS
179 013704 017737 166342 002274          MOV      @RLBA,B.BA      ;READ BA
180 013712 017737 166336 002276          MOV      @RLDA,B.DA      ;READ DA
181 013720 017737 166332 002300          MOV      @RLMP,B.MP      ;READ MP
182 013726 022737 000002 002410          CMP      #2,T.CNTRL      ;IF THE CONTOLLER IS AN RLV12
183 013734 001003          BNE     1$                ;READ BE
184 013736 017737 166316 002302          MOV      @RLBE,B.BE
185
186 013744 000205          1$:   RTS     R5
187
188
189          ;LOAD REGISTERS AT ERROR
190          ;CALL:  JSR      R5,AFTER
191
192 013746 017737 166276 002306  AFTER:  MOV      @RLCS,E.CS      ;READ CS
193 013754 017737 166272 002310          MOV      @RLBA,E.BA      ;READ BA
194 013762 017737 166266 002312          MOV      @RLDA,E.DA      ;READ DA
195 013770 017737 166262 002314          MOV      @RLMP,E.MP      ;READ MP
196 013776 017737 166254 002316          MOV      @RLMP,E.MP1     ;READ MP SECOND WORD IN SILO
197 014004 017737 166246 002320          MOV      @RLMP,E.MP2     ;READ MP THIRD WORD IN SILO
198 014012 022737 000002 002410          CMP      #2,T.CNTRL      ;IF THE CONTROLLER IS AN RLV12
199 014020 001003          BNE     1$                ;READ BE
200 014022 017737 166232 002322          MOV      @RLBE,E.BE
201
202 014030 000205          1$:   RTS     R5
203
204
205
206          .SBTTL  ROUTINE TO CALCULATE CRC
207
208          ;ROUTINE WILL CALCULATE A CRC-16 CRC ON A WORD OF
209          ;1-16 BITS IN LENGTH, RESULT IS RETURNED IN "CALBCC"
210          :
211          :      CALL:  JSR      R5,SIMBCC
212          :      .WORD      .WORD      ;NUMBER OF BITS (1-16)
213          :      .WORD      .WORD      ;DATA FOR CRC CALCULATION
214          :      .WORD      .WORD      ;PREVIOUS OR STARTING CRC
  
```



```

215                                     :
216                                     :
217                                     :
218 014032 010046 SIMBCC: MOV      R0,-(SP)      :SAVE R0
219 014034 010146      MOV      R1,-(SP)      :SAVE R1
220 014036 010246      MOV      R2,-(SP)      :SAVE R2
221                                     :
222 014040 012537 002346      MOV      (R5)+,TEMP2      :GET NUMBER OF BITS
223 014044 012537 002350      MOV      (R5)+,TEMP3      :GET DATA FOR CRC CALCULATION
224 014050 012537 002352      MOV      (R5)+,TEMP4      :GET STARTING CRC
225                                     :
226 014054 005037 002342 1$: CLR      BCCFBK      :
227 014060 013700 002352      MOV      TEMP4,R0      :GET PREVIOUS CRC
228 014064 006037 002350      ROR      TEMP3      :ROTATE NEW DATA
229 014070 005500      ADC      R0      :MERGE NEW WITH OLD
230 014072 032700 000001      BIT      #1,R0      :BIT 0 SET
231 014076 001402      BEQ      2$      :IF NOT CONTINUE
232 014100 005137 002342      COM      BCCFBK      :
233 014104 013700 002336 2$: MOV      XPOLY,R0      :GET CRC POLYNOMIAL (CRC-16)
234 014110 005100      COM      R0      :COMPLIMENT POLYNOMIAL
235 014112 040037 002342      BIC      R0,BCCFBK
236 014116 000241      CLC      :CLEAR CARRY
237 014120 006037 002352      ROR      TEMP4
238 014124 013700 002342      MOV      BCCFBK,R0
239 014130 013701 002352      MOV      TEMP4,R1
240 014134 010102      MOV      R1,R2
241 014136 040100      BIC      R1,R0
242 014140 043702 002342      BIC      BCCFBK,R2
243 014144 050200      BIS      R2,R0
244 014146 043737 002336 002352      BIC      XPOLY,TEMP4
245 014154 050037 002352      BIS      R0,TEMP4
246 014160 005337 002346      DEC      TEMP2
247 014164 001333      BNE      1$
248 014166 013737 002352 002344      MOV      TEMP4,CALBCC
249                                     :
250 014174 012602      MOV      (SP)+,R2      :RESTORE REGISTERS FROM STACK
251 014176 012601      MOV      (SP)+,R1
252 014200 012600      MOV      (SP)+,R0
253                                     :
254 014202 000205      RTS      R5      :RETURN
255                                     :
256                                     :
257                                     :
258                                     :ROUTINE TO SET FLAG IF TRAP OCCURRED
259                                     :"TRPHAN" IS IN LOCATION 4.
260                                     :
261                                     :
262 014204 005237 002326 TRPHAN: INC      TRPFLG      :INDICATE TRAP
263 014210 000002      RTI      :RETURN
264                                     :
265 014212      BGNSRV
266                                     :
267 014212 005237 002330 INTSRV: INC      INTFLG      :INDICATE INTERRUPT
268                                     :
269 014216      ENDSRV
014216 L10020:
014216 000002      RTI
  
```

```

270
271      ;ROUTINE TO WAIT FOR DRIVE READY
272 014220 010146      WTD RDY: MOV      R1,-(SP)      ;SAVE R1
273 014222 012701 003720      MOV      #2000.,R1      ;TIME OUT OF 200 MILLISECONDS
274 014226 032777 000001 166014 1$: BIT      #DRDY,@RLCS      ;DRIVE READY?
275 014234 001022      BNE      2$          ;YES, EXIT
276 014236      DELAY      1          ;WAIT A WHILE
      014236 012727 000001      MOV      #1,(PC)+
      014242 000000      .WORD      0
      014244 013727 002116      MOV      LSDLY,(PC)+
      014250 000000      .WORD      0
      014252 005367 177772      DEC      -6(PC)
      014256 001375      BNE      -4
      014260 005367 177756      DEC      -22(PC)
      014264 001367      BNE      -20
277 014266 005301      DEC      R1          ;CHECK IF TIME UP
278 014270 001356      BNE      1$          ;NO, GO CHECK DRIVE READY
279
280      ERRDF      200.,DRTIM,ERR5 ;DRIVE READY DID NOT SET
      014272 104455      TRAP      CSERDF
      014274 000310      .WORD      200
      014276 004712      .WORD      DRTIM
      014300 010372      .WORD      ERR5
281
282 014302 012601      2$:  MOV      (SP)+,R1      ;RESTORE
283 014304 000205      RTS      R5          ;EXIT
284
285      ;ROUTINE TO WAIT FOR CONTROLLER READY
286 014306 010146      WTC RDY: MOV      R1,-(SP)      ;SAVE R1
287 014310 012701 017500      MOV      #8000.,R1      ;WAIT 800 MILLISECONDS
288 014314 032777 000200 165726 1$: BIT      #CRDY,@RLCS      ;CONTROLLER READY
289 014322 001025      BNE      2$          ;YES, EXIT
290 014324      DELAY      1          ;WAIT A WHILE
      014324 012727 000001      MOV      #1,(PC)+
      014330 000000      .WORD      0
      014332 013727 002116      MOV      LSDLY,(PC)+
      014336 000000      .WORD      0
      014340 005367 177772      DEC      -6(PC)
      014344 001375      BNE      -4
      014346 005367 177756      DEC      -22(PC)
      014352 001367      BNE      -20
291 014354 005301      DEC      R1          ;CHECK IF TIME UP
292 014356 001356      BNE      1$          ;NO GO BACK
293
294 014360 004537 013746      JSR      R5,AFTER      ;GET REGISTERS
295
296      ERRDF      100.,CRTIM,ERR6 ;CONTROLLER TIMED OUT
      014364 104455      TRAP      CSERDF
      014366 000144      .WORD      100
      014370 004665      .WORD      CRTIM
      014372 010404      .WORD      ERR6
297
298 014374 000402      BR      3$          ;EXIT
299
300 014376 004537 013746      2$:  JSR      R5,AFTER      ;GET REGISTERS
301 014402 012601      3$:  MOV      (SP)+,R1
302 014404 000205      RTS      R5          ;EXIT
  
```


303
304 014406
305
306

ENDMOD

1
2
3 014406
4 014406
5
6
7
8
9 014406
10
11
12 014406 005037 002326
13 014412
014412 012746 000340
014416 012746 014204
014422 013746 002340
014426 012746 000003
014432 104437
014434 062706 000010
14
15 014440 005777 165604
16 014444
014444 013700 002340
014450 104436
17 014452 005737 002326
18 014456 001407
19 014460 013737 002250 002362
20
21 014466
014466 104454
014470 000000
014472 004740
014474 010176
22 014476
014476 104406
23 014500
014500
014500 104401
24
25
26
27
28 014502
29
30
31 014502
32
33
34
35
36 014502
37
38 014502 005037 002326

```
.SBTTL **TEST 1** - RLCS ADDRESSABILITY
BGNTST ;****START OF TEST****
STARS
:*****
:TEST TO SEE IF WE CAN ADDRESS THE CONTROL
:AND STATUS REGISTER. IF WE TRAP WE WILL REPORT
:THE ERROR AND ABORT. AFTER THIS TEST WE ONLY KNOW
:THAT WE CAN ADDRESS THE REGISTER.
STARS
:*****
1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
2$: SETVEC ERRVEC,#TRPHAN,#340 ;SET TO CATCH TRAP
MOV #340,-(SP)
MOV #TRPHAN,-(SP)
MOV ERRVEC,-(SP)
MOV #3,-(SP)
TRAP C$SVEC
ADD #10,SP
TST @RLCS ;ADDRESS RLCS
CLRVEC ERRVEC ;RELEASE TRAP VECTOR
MOV ERRVEC,R0
TRAP C$CVEC
TST TRPFLG ;TRAP OCCURRED???
BEQ 3$ ;NO, IKAY PROCEED
MOV RLCS,GDDAT ;SET UP ERROR DATA
ERRSF 0,,EM1,ERR1 ;BUS TIMEOUT IN ADDRESSING RLCS
TRAP C$ERSF
.WORD 0
.WORD EM1
.WORD ERR1
3$: CKLOOP ;CHECK IF /FL:LOE IS SET
TRAP C$CLP1
ENDTST ;****END OF TEST****
L10021: TRAP C$SETST
```

24
25
26
27
28 014502
29
30
31 014502
32
33
34
35
36 014502
37
38 014502 005037 002326

```
.SBTTL **TEST 2** - RLBA ADDRESSABILITY
BGNTST ;****START OF TEST****
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
:*****
1$: CLR TRPFLG ;CLEAR TRAP OCCURANCE
```



```

39 014506          2$:  SETVEC  ERRVEC,#TRPHAN,#340 ;SET TO CATCH TRAP
   014506 012746 000340      MOV    #340,-(SP)
   014512 012746 014204      MOV    #TRPHAN,-(SP)
   014516 013746 002340      MOV    ERRVEC,-(SP)
   014522 012746 000003      MOV    #3,-(SP)
   014526 104437          TRAP   C$SVEC
   014530 062706 000010      ADD    #10,SP

40
41 014534 005777 165512      TST    @RLBA          ;ADDRESS RLBA
42 014540          CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
   014540 013700 002340      MOV    ERRVEC,RO
   014544 104436          TRAP   C$CVEC
43 014546 005737 002326      TST    TRPFLG        ;TRAP OCCURRED???
44 014552 001407          BEQ    3$            ;NO, CONTINUE
45 014554 013737 002252 002362  MOV    RLBA,GDDAT    ;SETUP ERROR DATA
46
47 014562          ERRSF   1,EM2,ERR1    ;BUS TIMEOUT IN ADDRESSING RLBA
   014562 104454          TRAP   C$ERSF
   014564 000001          .WORD  1
   014566 004765          .WORD  EM2
   014570 010176          .WORD  ERR1

48 014572          3$:  CKLOOP  ;CHECK IF /FL:LOE IS SET
   014572 104406          TRAP   C$CLP1
49 014574          ENDTST  ;****END OF TEST****
   014574          L10022:
   014574 104401          TRAP   C$SETST

```

```

50
51
52          .SBTTL  **TEST 3** - RLDA ADDRESSABILITY
53
54 014576          BGNTST  ;****START OF TEST****
55 014576          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
   ;*****

```

```

61
62
63 014576 005037 002326      1$:  CLR    TRPFLG        ;CLEAR TRAP OCCURANCE
64 014602          2$:  SETVEC  ERRVEC,#TRPHAN,#340 ;SET TO CATCH TRAP
   014602 012746 000340      MOV    #340,-(SP)
   014606 012746 014204      MOV    #TRPHAN,-(SP)
   014612 013746 002340      MOV    ERRVEC,-(SP)
   014616 012746 000003      MOV    #3,-(SP)
   014622 104437          TRAP   C$SVEC
   014624 062706 000010      ADD    #10,SP

65
66 014630 005777 165420      TST    @RLDA          ;ADDRESS RLDA
67 014634          CLRVEC  ERRVEC          ;RELEASE TRAP VECTOR
   014634 013700 002340      MOV    ERRVEC,RO
   014640 104436          TRAP   C$CVEC
68 014642 005737 002326      TST    TRPFLG        ;TRAP OCCURRED???
69 014646 001407          BEQ    3$            ;NO, CONTINUE
70

```

```

71 014650 013737 002254 002362      MOV      RLDA,GDDAT      ;SETUP ERROR INFO
72 014656                                ERRSF    2.,EM3,ERR1    ;BUS TIMEOUT IN ADDRESSING RLDA
    014656 104454          TRAP      C$ERSF
    014660 000002          .WORD    2
    014662 005012          .WORD    EM3
    014664 010176          .WORD    ERR1
73 014666                                3$:      CKLOOP          ;CHECK IF /FL:LOE IS SET
    014666 104406          TRAP      C$CLP1
74 014670                                ENDTST
    014670                                L10023:
    014670 104401          TRAP      C$ETST
  
```

.SBTTL **TEST 4** - RLMP ADDRESSABILITY

```

75
76
77
78
79 014672                                BGNTST          ;****START OF TEST****
80 014672                                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
    ;*****
  
```

```

86
87
88 014672 005037 002326          1$:      CLR      TRPFLG          ;CLEAR TRAP OCCURANCE
89 014676                                2$:      SETVEC   ERRVEC,#TRPHAN,#340 ;SET UP TO CATCH TRAP
    014676 012746 000340          MOV      #340,-(SP)
    014702 012746 014204          MOV      #TRPHAN,-(SP)
    014706 013746 002340          MOV      ERRVEC,-(SP)
    014712 012746 000003          MOV      #3,-(SP)
    014716 104437          TRAP      C$SVEC
    014720 062706 000010          ADD      #10,SP
  
```

```

90
91 014724 005777 165326          TST      @RLMP          ;ADDRESS RLMP
92 014730                                CLRVEC   ERRVEC          ;RELEASE TRAP VECTOR
    014730 013700 002340          MOV      ERRVEC,R0
    014734 104436          TRAP      C$CVEC
93 014736 005737 002326          TST      TRPFLG          ;TRAP OCCURRED???
94 014742 001407          BEQ      3$              ;NO, CONTINUE
95 014744 013737 002256 002362      MOV      RLMP,GDDAT      ;SET UP ERROR INFO
96
  
```

```

97 014752                                ERRSF    3.,EM4,ERR1    ;BUS TIMEOUT IN ADDRESSING RLMP
    014752 104454          TRAP      C$ERSF
    014754 000003          .WORD    3
    014756 005037          .WORD    EM4
    014760 010176          .WORD    ERR1
98 014762                                3$:      CKLOOP          ;CHECK IF /FL:LOE IS SET
    014762 104406          TRAP      C$CLP1
99 014764                                ENDTST
    014764                                L10024:
    014764 104401          TRAP      C$ETST
  
```

```

100
101
102
103
.SBTTL **TEST 5** - READ WRITE OF RLCS
  
```



```

104 014766          BGNTST          ;****START OF TEST****
105
106
107
108 014766          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
:*****
109
110
111
112
113 014766
114
115
116 014766 012703 002772          MOV      #CSPAT,R3          ;SET UP TABLE POINTER OF PATTERNS
117
118 014772          BGNSEG          ;****START OF SEGMENT****
014772 104404          TRAP      CSBSEG
119
120 014774          CTEST:
121 014774 011337 002362          MOV      (R3),GDDAT        ;GET PATTERN INTO GDDAT
122 015000 052737 000200 002362  BIS      #200,GDDAT        ;INSURE GO IS SET
123 015006 013777 002362 165234  MOV      GDDAT,@RLCS      ;LOAD RLCS (CONTROL AND STATUS)
124 015014 032777 040000 165226  BIT      #DERR,@RLCS      ;IF DRIVE ERROR PRESENT
125 015022 001403          BEQ      99$              ;THEN EXPECT DRIVE AND
126 015024 052737 140000 002362  BIS      #ERR!DERR,GDDAT   ;COMPOSITE ERROR
127 015032 017737 165212 002364  99$:    MOV      @RLCS,BDDAT      ;READ RLCS BACK
128 015040 042737 000001 002364  BIC      #DRDY,BDDAT      ;IGNORE DRIVE READY
129 015046 023737 002362 002364  CMP      GDDAT,BDDAT      ;DID WE READ WHAT WE LOADED
130 015054 001404          BEQ      1$              ;YES, THEN BRANCH
131
132 015056          ERRDF      4,,EM5,ERR2          ;WRONG DATA IN RLCS
015056 104455          TRAP      C$ERDF
015060 000004          .WORD     4
015062 005064          .WORD     EM5
015064 010210          .WORD     ERR2
133 015066          1$:    ESCAPE   SEG          ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
015066 104410          TRAP      C$ESCAPE
015070 000012          .WORD     10000$-.
134
135
136 015072 005723          TST      (R3)+            ;BUMP FOR NEXT PATTERN
137 015074 020327 003070          CMP      R3,#CSEND        ;CHECK FOR END
138 015100 001335          BNE      C$TEST          ;NOT END, LOAD NEXT PATTERN
139
140 015102          ENDSEG          ;****END OF SEGMENT****
015102 10000$:
141 015104          TRAP      C$ESEG
015104 104405          ;****END OF TEST****
142 015104          ENDTST
015104 104401          L10025:
TRAP      C$ETST
143
144          .SBTTL  **TEST 6** - READ WRITE OF RLBA
145
146 015106          BGNTST          ;****START OF TEST****
147

```

148 015106

STARS

 :TEST THAT WE CAN WRITE/READ BITS 15-1 OF THE
 :BUS ADDRESS REGISTER. FOUR PATTERNS ARE USED: GROWING 1, SHIFTING 1,
 :GROWING 0 AND SHIFTING 0. BIT 0 IS ALSO LOADED BUT
 :SHOULD ALWAYS COME BACK AS 0

149

150

151

152

153 015106

STARS

154

155

156 015106 012703 002416

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

157 015112

015112 104404

TRAP C\$BSEG

158 015114

BATEST:

159 015114 011337 002362

MOV (R3),GDDAT :GET PATTERN TO SEND

160 015120 022737 000001 002410

CMP #1,T.CNTRL ;RL11??

161 015126 002403

BLT 2\$:NO

162 015130 042737 000001 002362

BIC #BIT0,GDDAT :KEEP RLBA EVEN (UNIBUS)

163 015136 013777 002362 165106

2\$: MOV GDDAT,@RLBA :LOAD PATTERN TO BUS ADDRESS

164 015144 017737 165102 002364

MOV @RLBA,BDDAT :READ IT BACK

165 015152 023737 002362 002364

CMP GDDAT,BDDAT :IS IT CORRECT?

166 015160 001404

BEQ 1\$:IF SO, BRANCH

167

168 015162

ERRDF 5,EM6,ERR2 :DATA WRONG IN RLBA

015162 104455

TRAP C\$ERRDF

015164 000005

.WORD 5

015166 005135

.WORD EM6

015170 010210

.WORD ERR2

169 015172

1\$: ESCAPE SEG :IF /FL:LOE SET LOOP, ELSE EXIT SEG

015172 104410

TRAP C\$ESCAPE

015174 000012

.WORD 10000\$-


```

1
2
3 015176 005723          TST      (R3)+      :BUMP FOR NEXT PATTERN
4 015200 020327 002624  CMP      R3,#ENDPAT  :CHECK FOR END
5 015204 001343          BNE      BATEST     :NOT END, BRANCH FOR NEXT
6
7 015206          ENDSEG          :****END OF SEGMENT****
  015206          10000$:
  015206 104405      TRAP      C$ESEG
8 015210          ENDTST          :****END OF TEST****
  015210          L10026:
  015210 104401      TRAP      C$ETST
9
10
11      .SBTTL  **TEST 7** - READ WRITE OF RLDA
12
13 015212          BGNTST          :****START OF TEST****
14
15 015212          STARS
16          :*****
17          :TEST THAT WE CAN WRITE/READ THE DISK ADDRESS REGISTER
18          :ALL BIT POSITIONS ARE WRITTEN USING FOUR PATTERNS:
19 015212          :GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
          STARS
          :*****
20
21
22 015212 012703 002416  BGNSEG  MOV      #BEGPAT,R3  :SET UP POINTER TO PATTERN LIST
23 015216          TRAP      C$BSEG  :****START OF SEGMENT****
  015216 104404
24 015220          DATEST:
25 015220 011337 002362  MOV      (R3),GDDAT  :GET PATTERN
26 015224 013777 002362  MOV      GDDAT,@RLDA :LOAD PATTERN IN DA
  165022
27
28 015232 017737 165016 002364  MOV      @RLDA,BDDAT  :READ PATTERN BACK
29 015240 023737 002362 002364  CMP      GDDAT,BDDAT  :IS IT CORRECT?
30 015246 001404          BEQ      1$           :BRANCH IF CORRECT
31
32 015250          ERRDF      6,EM7,ERR2 :WRONG DATA IN RLDA
  015250 104455          TRAP      C$ERRDF
  015252 000006          .WORD      6
  015254 005163          .WORD      EM7
  015256 010210          .WORD      ERR2
33 015260          1$:      ESCAPE  SEG           :IF /FL:LOE SET LOOP, ELSE EXIT SEG
  015260 104410          TRAP      C$ESCAPE
  015262 000012          .WORD      10000$-.
34
35
36 015264 005723          TST      (R3)+      :BUMP POINTER
37 015266 020327 002624  CMP      R3,#ENDPAT  :AT END OF PATTERNS?
38 015272 001352          BNE      DATEST     :NO, BRANCH BACK
39
40 015274          ENDSEG          :****END OF SEGMENT****
  015274          10000$:
  015274 104405      TRAP      C$ESEG
41 015276          ENDTST          :****END OF TEST****
  015276          L10027:
  
```

```

015276 104401          TRAP  C$ETST
42
43
44          .SBTTL  **TEST 8** - BIS OF RLCS
45
46 015300          BGNTST          ;****START OF TEST****
47 015300          STARS
;*****
;TEST THAT WE CAN USE THE 'BIS' INSTRUCTION ON THE CONTROL
;AND STATUS REGISTER. BITS 8,9 AND 6-1 ARE TESTED TO
;SET INDIVIDUALLY AS WELL AS COLLECTIVELY WITHOUT DESTROYING
;ANY PREVIOUS DATA PATTERN
;*****
52 015300          STARS
;*****
53
54
55 015300 012703 002772          MOV    #CSPAT,R3          ;GET BEGINNING OF LIST
56 015304          BGNSEG          ;****START OF SEGMENT****
015304 104404          TRAP  C$BSEG
57 015306          1$:
58 015306 012777 000200 164734          MOV    #CRDY,@RLCS          ;INSURE GO IS THERE
59 015314 011337 002362          MOV    (R3),GDDAT          ;SET UP EXPECTED RLCS
60 015320 052737 000200 002362          BIS    #CRDY,GDDAT          ;IN GDDAT
61 015326 051377 164716          BIS    (R3),@RLCS          ;BIT SET PATTERN IN RLCS
62 015332 032777 040000 164710          BIT    #DERR,@RLCS          ;IF ERROR BIT SET THEN
63 015340 001403          BEQ    99$          ;EXPECT IT ON THE READ
64 015342 052737 140000 002362          BIS    #ERR!DERR,GDDAT          ;BACK
65 015350 017737 164674 002364 99$:          MOV    @RLCS,BDDAT          ;READ RLCS TO CHECK 'BIS'
66 015356 042737 000001 002364          BIC    #DRDY,BDDAT          ;CLEAR OUT DRIVE READY
67 015364 023737 002364 002362          CMP    BDDAT,GDDAT          ;DID BIS WORK?
68 015372 001404          BEQ    2$          ;BRANCH IF OKAY
69
70 015374          ERRDF  7.,EM61,ERR2          ;WRONG DATA IN RLCS
015374 104455          TRAP  C$ERDF
015376 000007          .WORD  7
015400 006660          .WORD  EM61
015402 010210          .WORD  ERR2
71 015404          2$:          ESCAPE SEG          ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
015404 104410          TRAP  C$ESCAPE
015406 000012          .WORD  10000$-.          ;BIT OR CLEARED OTHER BIT
72
73
74 015410 005723          TST    (R3)+          ;GET NEXT PATTERN
75 015412 022703 003070          CMP    #CSEND,R3          ;AT END OF LIST
76 015416 001333          BNE    1$          ;NO GO BACK FOR TEST OF
77          ;NEXT PATTERN
78 015420          ENDSEG          ;****END OF SEGMENT****
015420          10000$:
79 015422          TRAP  C$ESEG          ;****END OF TEST****
015422          ENDTST
015422          L10030:
015422 104401          TRAP  C$ETST
80
81
82          .SBTTL  **TEST 9** - BIC OF RLCS
83
84 015424          BGNTST          ;****START OF TEST****

```



```

85
86 015424
87
88
89
90 015424
91
92
93 015424 012703 002772
94 015430 104404
95 015432
96 015432 012777 001776 164610
97 015440 012737 001776 002362
98 015446 041337 002362
99 015452 041377 164572
100 015456 032777 040000 164564
101 015464 001403
102 015466 052737 140000 002362
103 015474 017737 164550 002364
104 015502 042737 000001 002364
105 015510 023737 002364 002362
106 015516 001404
107
108 015520
    015520 104455
    015522 000010
    015524 006741
    015526 010210
109 015530
    015530 104410
    015532 000012
110
111 015534 005723
112 015536 020327 003070
113 015542 001333
114 015544
    015544
    015544 104405
115 015546
    015546
    015546 104401
116
117
118
119
120 015550
121
122 015550
123
124
125
126
127 015550
  
```

```

STARS
:*****
:TEST THAT THE 'BIC' INSTRUCTION WILL WORK ON THE
:CONTROL AND STATUS REGISTER. BITS 8-9 AND 6-1 ARE
:TESTED.
STARS
:*****

BGNSEG MOV #CSPAT,R3 :GET BEGINNING OF PATTERNS
:*****START OF SEGMENT****

1$: TRAP C$SEEG

MOV #1776,@RLCS :SET ALL SETTABLE BITS
MOV #1776,GDDAT :SET UP EXPECT DATA IN
BIC (R3),GDDAT :GDDAT
BIC (R3),@RLCS :CLEAR BITS IN RLCS VIA 'BIC'
BIT #DERR,@RLCS :IF DRIVE ERROR BIT SET
BEQ 99$ :EXPECT IT SET WHEN WE
BIS #ERR!DERR,GDDAT :READ IT BACK
99$: MOV @RLCS,BDDAT :MOVE RLCS TO BDDAT FOR COMPARE
BIC #DRDY,BDDAT :CLEAR DRIVE READY
CMP BDDAT,GDDAT :DID 'BIC' WORK PROPERLY
BEQ 2$ :BRANCH IF OKAY

ERRDF 8.,EM62,ERR2 :WRONG DATA IN RLCS..
TRAP C$ERDF
.WORD 8
.WORD EM62
.WORD ERR2
2$: ESCAPE SEG :IF /FL:LOE SET LOOP, ELSE EXIT SEG
TRAP C$ESCAPE
.WORD 10000$-.

TST (R3)+ :GET NEXT PATTERN
CMP R3,#CSEND :AT END OF LIST
BNE 1$ :NO, GO BACK WITH NEXT PATTERN
:*****END OF SEGMENT****

ENDSEG 10000$:
TRAP C$ESEG :*****END OF TEST****

ENDTST L10031:
TRAP C$ETST

.SBTTL **TEST 10** - BIS OF RLBA
BGNST :*****START OF TEST****

STARS
:*****
:TEST THAT THE 'BIS' INSTRUCTION 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
  
```

```

  128
  129
  130 015550 012703 002416
  131 015554 104404
  132 015556
  133 015556 005077 164470
  134 015562 011337 002362
  135 015566 022737 000001 002410
  136 015574 002403
  137 015576 042737 000001 002362
  138 015604 051377 164442
  139 015610 017737 164436 002364
  140 015616 023737 002364 002362
  141 015624 001404
  142
  143 015626
  144 015636
  145
  146 015642 005723
  147 015644 020327 002624
  148 015650 001342
  149 015652
  150 015654
  151
  152
  153
  154
  155 015656
  156
  157 015656
  158
  159
  160
  161 015656
  162
  163
  164 015656 012703 002416
  165 015662
  166 015664
  167 015664 012777 177776 164360
  168 015672 012737 177776 002362
  169 015700 041337 002362
  
```

```

  ;*****
  BGNSEG MOV #BEGPAT,R3 ;GET START OF LIST
  TRAP C$BSEG ;****START OF SEGMENT****
  1$: CLR @RLBA ;CLEAR 'BA'
  MOV (R3),GDDAT ;SET EXPECTED
  CMP #1,T.CNTRL ;RL11
  BLT 3$ ;NO
  BIC #1,GDDAT ;BIT 0 CAN'T SET IN RLBA (UNIBUS)
  3$: BIS (R3),@RLBA ;BIS RLBA WITH PATTERN
  MOV @RLBA,BDDAT ;READ 'BA'
  CMP BDDAT,GDDAT ;DID RLBA LOAD PROPERLY?
  BEQ 2$ ;BRANCH IF YES
  ERRDF 9,EM63,ERR2 ;WRONG DATA IN RLBA
  TRAP C$ERDF
  .WORD 9
  .WORD EM63
  .WORD ERR2
  2$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
  TRAP C$ESCAPE
  .WORD 10000$-
  TST (R3)+ ;GET NEXT PATTERN
  CMP R3,#ENDPAT ;DID WE COMPLETE LIST
  BNE 1$ ;NO, GO BACK FOR NEXT.
  ENDSEG ;****END OF SEGMENT****
  10000$: TRAP C$ESEG
  ENDTST ;****END OF TEST****
  L10032: TRAP C$ETST
  
```

```

  .SBTTL **TEST 11** - BIC OF RLBA
  BGNTST ;****START OF TEST****
  STARS
  ;*****
  ;TEST THAT THE 'BIC' INSTRUCTION WILL WORK ON THE BUS
  ;ADDRESS REGISTER. BITS 15-1 ARE TESTED WITH 4 PATTERNS
  ;GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0.
  STARS
  ;*****
  
```

```

  BGNSEG MOV #BEGPAT,R3 ;GET START OF LIST
  TRAP C$BSEG ;****START OF SEGMENT****
  1$: MOV #-2,@RLBA ;SET RLBA TO ALL 1'S (BIT 0=0)
  MOV #-2,GDDAT ;SET UP EXPECTED RESULTS
  BIC (R3),GDDAT ;IN GDDAT
  
```



```

170 015704 041377 164342          BIC      (R3),@RLBA      :BIC RLBA
171 015710 017737 164336 002364  MOV      @RLBA,BDDAT   :READ RLBA
172 015716 023737 002364 002362  CMP      BDDAT,GDDAT   :BIC WORK OKAY?
173 015724 001404                      BEQ      2$            :IF YES BRANCH
174
175 015726                      ERRDF   10.,EM64,ERR2   :WRONG DATA IN RLBA
    015726 104455          TRAP    C$ERRDF
    015730 000012          .WORD   10
    015732 007105          .WORD   EM64
    015734 010210          .WORD   ERR2
176 015736                      2$:    ESCAPE  SEG          :IF /FL:LOE SET LOOP, ELSE EXIT SEG
    015736 104410          TRAP    C$ESCAPE
    015740 000012          .WORD  10000$-.
177
178 015742 005723          TST     (R3)+         :GET NEXT PATTERN
179 015744 020327 002624  CMP     R3,#ENDPAT    :HAVE WE COMPLETED LIST
180 015750 001345          BNE     1$            :NO, GO BACK FOR NEXT
181 015752          ENDSEG 10000$:          :****END OF SEGMENT****
    015752 104405          TRAP    C$ESEG
182 015754          ENDTST L10033:          :****END OF TEST****
    015754 104401          TRAP    C$ETST
  
```

.SBTTL **TEST 12** - BIS OF RLDA

```

183
184
185
186
187 015756          BGNTST          :****START OF TEST****
188
189 015756          STARS
    :*****
    :TEST THAT THE 'BIS' INSTRUCTION WILL WORK ON THE DISK ADDRESS
    :REGISTER. BITS 15-0 ARE TESTED WITH 4 PATTERNS, GROWING 1,
    :SHIFTING 1, GROWING 0, AND SHIFTING 0.
    :*****
190
191
192
193 015756
194
195
196 015756 012703 002416          MOV     #BEGPAT,R3    :GET START OF LIST
197 015762          BGNSEG 1$:    TRAP    C$BSEG      :****START OF SEGMENT****
    015762 104404
198 015764
199 015764 005077 164264          CLR     @RLDA         :CLEAR 'DA'
200 015770 011337 002362          MOV     (R3),GDDAT    :SET EXPECTED
201 015774 051377 164254          BIS     (R3),@RLDA    :BIS RLDA
202 016000 017737 164250 002364  MOV     @RLDA,BDDAT   :READ RLDA
203 016006 023737 002364 002362  CMP     BDDAT,GDDAT   :IS RLDA CORRECT
204 016014 001404                      BEQ     2$            :IF OKAY BRANCH
205
206 016016                      ERRDF   11.,EM65,ERR2   :WRONG DATA IN RLDA
    016016 104455          TRAP    C$ERRDF
    016020 000013          .WORD   11
    016022 007170          .WORD   EM65
    016024 010210          .WORD   ERR2
207 016026                      2$:    ESCAPE  SEG          :IF /FL:LOE SET LOOP, ELSE EXIT SEG
    016026 104410          TRAP    C$ESCAPE
    016030 000012          .WORD  10000$-.
  
```

```

208
209 016032 005723          TST      (R3)+      :GET NEXT PATTERN
210 016034 020327 002624  CMP      R3,#ENDPAT :HAVE WE FINISHED?
211 016040 001351          BNE      1$         :NO GO BACK
212 016042          ENDSEG          :****END OF SEGMENT****
      016042          10000$:
213 016042 104405          TRAP     C$ESEG
      016044          ENDTST          :****END OF TEST****
      016044          L10034:
      016044 104401          TRAP     C$ETST

214
215
216          .SBTTL  **TEST 13** - BIC OF RLDA
217
218 016046          BGNTST          :****START OF TEST****
219
220 016046          STARS
      :*****
      :TEST THAT THE 'BIC' INSTRUCTION WORKS ON THE DISK
      :ADDRESS REGISTER. ALL BITS ARE TESTED WITH FOUR
      :PATTERNS: GROWING 1, SHIFTING 1, GROWING 0 AND SHIFTING 0
      STARS
      :*****
221
222
223
224 016046
225
226
227 016046 012703 002416  BGNSEG  MOV      #BEGPAT,R3  :GET START OF LIST
228 016052          TRAP     C$BSEG  :****START OF SEGMENT****
      016052 104404          1$:
229 016054          MOV      #-1,@RLDA  :SET RLDA TO ALL 1'S
230 016054 012777 177777 164172  MOV      #-1,GDDAT  :SET EXPECTED DATA
231 016062 012737 177777 002362  BIC      (R3),GDDAT  :SET EXPECTED DATA
232 016070 041337 002362          BIC      (R3),@RLDA  :'BIC' RLDA
233 016074 041377 164154          MOV      @RLDA,BDDAT :READ RLDA
234 016100 017737 164150 002364  CMP      GDDAT,BDDAT :DID 'BIC' WORK?
235 016106 023737 002362 002364  BEQ      2$         :IF IT DID BRANCH
236 016114 001404
237
238 016116          ERRDF   12,EM66,ERR2 :WRONG DATA IN RLDA
      016116 104455          TRAP     C$ERDF
      016120 000014          .WORD   12
      016122 007251          .WORD   EM66
      016124 010210          .WORD   ERR2
239 016126          ESCAPE  SEG          :IF /FL:LOE SET LOOP, ELSE EXIT SEG
      016126 104410          TRAP     C$ESCAPE
      016130 000012          .WORD   10000$-.

240
241 016132 005723          TST      (R3)+      :GET NEXT PATTERN
242 016134 020327 002624  CMP      R3,#ENDPAT :DONE?
243 016140 001345          BNE      1$         :NO GO BACK
244 016142          ENDSEG          :****END OF SEGMENT****
      016142          10000$:
245 016142 104405          TRAP     C$ESEG
      016144          ENDTST          :****END OF TEST****
      016144          L10035:
      016144 104401          TRAP     C$ETST

246
247
  
```


248
249
250 016146
251
252 016146

253
254
255
256
257
258
259
260
261 016146

262
263
264 016146 012700 000340
016146 104441
265 016154 012777 000377 164066
266 016162 012737 000200 002362
267 016170 032777 040000 164052
268 016176 001403
269 016200 052737 140000 002362
270 016206 012700 000100
271 016212
016212 104433
272 016214 005300
273 016216 001376
274 016220 017737 164024 002364
275 016226 042737 000001 002364
276 016234 023737 002364 002362
277 016242 001404
278
279 016244
016244 104455
016246 000015
016250 007334
016252 010210
280 016254
281 016254
016254
016254 104401

282
283
284
285
286 016256
287
288 016256

289
290
291
292 016256

```
.SBTTL **TEST 14** - BUS RESET OF RLCS
BGNTST ;****START OF TEST****
STARS
:*****
:TEST THAT A BUS RESET WILL CLEAR THE PROPER BITS
:OF THE CONTROL AND STATUS REGISTER. THOSE BITS ARE
:BITS 6-1,8,9,10,11,12,13,15. 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.
STARS
:*****
SETPRI #PRI07 ;PRIORITY TO SEVEN
MOV #PRI07,R0
TRAP C$SPRI
MOV #377,@RLCS ;LOAD ALL RLCS LOADABLE BITS
MOV #CRDY,GDDAT ;SETUP EXPECTED
BIT #DERR,@RLCS ;DRIVE ERR SET?
BEQ 1$ ;IF NOT DON'T EXPECT IT
BIS #DERR!ERR,GDDAT ;IT'S SET, INIT BETTER NOT CLR
1$: MOV #100,R0 ;SET UP A WAIT LOOP
BRESET ;BUS RESET
TRAP C$RESET
2$: DEC R0 ;WAIT IN CASE OF DRIVE ERROR
BNE 2$
MOV @RLCS,BDDAT ;READ RLCS
BIC #DRDY,BDDAT ;CLEAR OUT DRDY - DON'T CARE
CMP BDDAT,GDDAT ;DID INIT WORK
BEQ 3$ ;YES, BRANCH
ERRDF 13.,EM67,ERR2 ;WRONG DATA IN RLCS
TRAP C$ERDF
.WORD 13
.WORD EM67
.WORD ERR2
3$:
ENDTST ;****END OF TEST****
L10036: TRAP C$SETST
```

```
.SBTTL **TEST 15** - BUS RESET OF RLBA
BGNTST ;****START OF TEST****
STARS
:*****
:TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
:BUS ADDRESS REGISTER. THE BUS ADDRESS IS LOADED WITH 177776
:AND IS EXPECTED TO BE ZERO AFTER THE RESET
STARS
```

```

;:*****
293
294
295 016256 012777 177776 163766      MOV    #-2,@RLBA      ;SET BA TO ALL 1'S
296 016264 022737 000001 002410      CMP    #1,T.CNTRL    ;RL11??
297 016272 002403                    BLT    2$            ;NO
298 016274 052777 000001 163750      BIS    #1,@RLBA
299 016302 005037 002362      2$:  CLR    GDDAT      ;CLEAR EXPECTED DATA
300 016306                    BRESET ;ISSUE BUS INIT
      016306 104433      TRAP   C$RESET
301 016310 017737 163736 002364      MOV    @RLBA,BDDAT  ;READ RLBA
302 016316 001404                    BEQ    1$            ;IF CLEAR BRANCH
303
304 016320                    ERRDF  14.,EM70,ERR2 ;WRONG DATA IN RLBA
      016320 104455      TRAP   C$ERDF
      016322 000016      .WORD 14
      016324 007371      .WORD EM70
      016326 010210      .WORD ERR2
305 016330      1$:
306
307 016330      ENDTST ;****END OF TEST****
      016330
      016330 104401      L10037: TRAP   C$ETST
308
309
310      .SBTTL **TEST 16** - BUS RESET OF RLDA
311
312 016332      BGNTST ;****START OF TEST****
313
314 016332      STARS
;:*****
;:TEST THAT A BUS RESET WILL CLEAR THE ENTIRE
;:DISK ADDRESS REGISTER. THE DISK ADDRESS IS LOADED WITH 177777
;:AND IS EXPECTED TO BE ZERO AFTER THE RESET.
;:STARS
;:*****
315
316
317
318 016332
319
320
321 016332 012777 177777 163714      MOV    #-1,@RLDA    ;SET DA TO ALL 1'S
322 016340 005037 002362      CLR    GDDAT      ;CLEAR EXPECTED
323 016344                    BRESET ;ISSUE BUS INIT
      016344 104433      TRAP   C$RESET
324 016346 017737 163702 002364      MOV    @RLDA,BDDAT ;READ RLDA
325 016354 001404                    BEQ    1$            ;IF CLEAR BRANCH
326
327 016356                    ERRDF  15.,EM71,ERR2 ;WRONG DATA IN RLDA
      016356 104455      TRAP   C$ERDF
      016360 000017      .WORD 15
      016362 007426      .WORD EM71
      016364 010210      .WORD ERR2
328 016366      1$:
329
330 016366      ENDTST ;****END OF TEST****
      016366
      016366 104401      L10040: TRAP   C$ETST
331
332
  
```



```

333      .SBTTL  **TEST 17** - UNIQUENESS OF RLCS
334
335 016370      BGNTST      ;****START OF TEST****
336
337 016370      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.
338      STARS
339      :*****
340
341
342
343 016370
344
345
346 016370 012737 000201 002332      MOV      #DRDY!CRDY,LDCSR      ;SET DRIVE AND CONTROLLER READY
347 016376 012777 177776 163646      MOV      #-2,@RLBA      ;SET RLBA TO ALL 1'S
348 016404 012777 177777 163642      MOV      #-1,@RLDA      ;SET RLDA TO ALL 1'S
349 016412 013777 002332 163630      MOV      LDCSR,@RLCS      ;WRITE RLCS
350
351      ;CHECK THAT RLBA REMAINS UNAFFECTED
352
353 016420 022777 177776 163624      CMP      #-2,@RLBA      ;RLBA OKAY?
354 016426 001412      BEQ      1$      ;YES, GO CHECK DA
355
356 016430 012737 177776 002362      MOV      #-2,GDDAT      ;SET UP EXPECTED
357 016436 017737 163610 002364      MOV      @RLBA,BDDAT      ;READ RLBA
358
359 016444      ERRDF      16.,EM72,ERR2      ;CS MODIFIED BA
   016444 104455      TRAP      C$ERDF
   016446 000020      .WORD      16
   016450 007463      .WORD      EM72
   016452 010210      .WORD      ERR2
360 016454      1$:      CKLOOP      ;CHECK IF /FL:LOE IS SET
   016454 104406      TRAP      C$CLP1
361
362 016456 022777 177777 163570      CMP      #-1,@RLDA      ;RLDA OKAY?
363 016464 001412      BEQ      2$      ;YES, CONTINUE
364
365 016466 012737 177777 002362      MOV      #-1,GDDAT      ;SET UP EXPECTED
366 016474 017737 163554 002364      MOV      @RLDA,BDDAT      ;READ DA
367
368 016502      ERRDF      17.,EM73,ERR2      ;CS MODIFIED DA
   016502 104455      TRAP      C$ERDF
   016504 000021      .WORD      17
   016506 007516      .WORD      EM73
   016510 010210      .WORD      ERR2
369 016512      2$:
370
371
372 016512      ENDTST      ;****END OF TEST****
   016512      L10041:
   016512 104401      TRAP      C$ETST
373
374
375      .SBTTL  **TEST 18** - UNIQUENESS OF RLBA
376
  
```

377 016514
 378 016514
 379
 380
 381
 382
 383
 384 016514

BGNTST ;****START OF TEST****
 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
 :BY WRITING THE RLBA.
 STARS
 :*****

385
 386
 387 016514 012737 000200 002362
 388 016522 032777 040000 163520
 389 016530 001403
 390 016532 052737 140000 002362
 391 016540 013777 002362 163502
 392 016546 012777 177777 163500
 393 016554 005077 163472

MOV #CRDY,GDDAT ;CONTROLLER READY
 BIT #DERR,@RLCS ;IF DRIVE ERROR IS
 BEQ 99\$;SET THEN EXPECT IT
 BIS #ERR!DERR,GDDAT ;SET WHEN WE READ IT.
 99\$: MOV GDDAT,@RLCS ;LOAD RLCS
 MOV #-1,@RLDA ;LOAD RLDA
 CLR @RLBA ;CLEAR RLBA

394
 395
 396
 397 016560 017737 163464 002364
 398 016566 042737 000001 002364
 399 016574 023737 002364 002362
 400 016602 001404

;CHECK IF RLCS IS OKAY
 MOV @RLCS,BDDAT ;READ RLCS
 BIC #DRDY,BDDAT ;IGNORE DRIVE READY
 CMP BDDAT,GDDAT ;CS OK?
 BEQ 1\$;YES, GO CHECK DA

401
 402 016604
 016604 104455
 016606 000022
 016610 007551
 016612 010210
 403 016614
 016614 104406

ERRDF 18.,EM74,ERR2 ;BA MODIFIED CS
 TRAP C\$ERDF
 .WORD 18
 .WORD EM74
 .WORD ERR2
 1\$: CKLOOP ;CHECK IF /FL:LOE IS SET
 TRAP C\$CLP1

404
 405 016616 022777 177777 163430
 406
 407 016624 001412
 408
 409 016626 012737 177777 002362
 410 016634 017737 163414 002364

CMP #-1,@RLDA ;IS RLDA OKAY?
 BEQ 2\$;IF OKAY BRANCH
 MOV #-1,GDDAT ;SET UP EXPECTED
 MOV @RLDA,BDDAT ;READ RLDA

411
 412 016642
 016642 104455
 016644 000023
 016646 007603
 016650 010210

ERRDF 19.,EM75,ERR2 ;BA MODIFIED DA
 TRAP C\$ERDF
 .WORD 19
 .WORD EM75
 .WORD ERR2

413 016652
 414 016652
 016652
 016652 104401

2\$:
 ENDTST ;****END OF TEST****
 L10042:
 TRAP C\$SETST

415
 416
 417
 418
 419 016654
 420

.SBTTL **TEST 19** - UNIQUENESS OF RLDA
 BGNTST ;****START OF TEST****


```

421
422 016654          STARS
                    :*****
423                :TEST THE UNIQUENESS OF THE DISK ADDRESS REGISTER.  THE RLCS
424                :AND RLBA ARE LOADED WITH XXX20X AND 177776
425                :RESPECTIVELY.  THE RLDA IS THEN WRITTEN TO INSURE
426                :THAT NEITHER THE RLCS OR THE RLBA ARE MODIFIED
427                :BY WRITING THE RLDA.
428 016654          STARS
                    :*****
429
430
431 016654 012737 000200 002362          MOV      #CRDY,GDDAT      :CONTROLLER READY
432 016662 032777 040000 163360          BIT      #DERR,@RLCS    :IF DRIVE ERROR SET
433 016670 001403                    BEQ      99$              :THEN EXPECT IT LATER
434 016672 052737 140000 002362          BIS      #ERR!DERR,GDDAT
435 016700 013777 002362 163342 99$:    MOV      GDDAT,@RLCS    :LOAD CS
436 016706 012777 177776 163336          MOV      #-2,@RLBA     :LOAD BA WITH ALL 1'S
437 016714 005077 163334                    CLR      @RLDA          :CLEAR RLDA
438
439                :CHECK IF RLCS IS OKAY
440
441 016720 017737 163324 002364          MOV      @RLCS,BDDAT   :READ RLCS
442 016726 042737 000001 002364          BIC      #DRDY,BDDAT   :IGNORE DRIVE READY
443 016734 023737 002362 002364          CMP      GDDAT,BDDAT   :RLCS OKAY?
444 016742 001404                    BEQ      1$              :YES, THEN BRANCH
445
446 016744                    ERRDF  20.,EM76,ERR2  :DA MODIFIED CS
447 016744 104455                    TRAP   C$ERDF
448 016746 000024                    .WORD  20
449 016750 007635                    .WORD  EM76
450 016752 010210                    .WORD  ERR2
451 016754                    1$:    CKLOOP
452 016754 104406                    TRAP   C$CLP1           :CHECK IF /FL:LOE IS SET
453
454 016756 022777 177776 163266          CMP      #-2,@RLBA     :IS RLBA OKAY?
455 016764 001412                    BEQ      2$              :BRANCH IF OKAY
456
457 016766 012737 177776 002362          MOV      #-2,GDDAT     :SET UP EXPECTED
458 016774 017737 163252 002364          MOV      @RLBA,BDDAT   :READ RLBA
459
460 017002                    ERRDF  21.,EM77,ERR2  :DA MODIFIED BA
461 017002 104455                    TRAP   C$ERDF
462 017004 000025                    .WORD  21
463 017006 007670                    .WORD  EM77
464 017010 010210                    .WORD  ERR2
465
466 017012                    2$:
467
468
469 017012                    ENDTST
470 017012                    L10043:
471 017012 104401                    TRAP   C$SETST
472
473                .SBTTL  **TEST 20** - UNIQUENESS OF RLMP
474
475                BGNTST
476
477                :*****START OF TEST****

```

465
 466 017014

467
 468
 469
 470
 471 017014

STARS
 :*****
 :TEST THE UNIQUENESS OF THE MULTI-PURPOSE REGISTER
 :WE WILL WRITE THE RLCS, RLBA, AND THE RLDA, THEN THE
 :RLMP IS WRITTEN. WE THEN GO BACK AN VERIFY THE CONTENTS
 :OF THE RLCS, RLBA, RLDA.
 STARS
 :*****

472
 473
 474 017014 012737 000200 002362
 475 017022 032777 040000 163220
 476 017030 001403
 477 017032 052737 140000 002362
 478 017040 013777 002362 163202
 479 017046 012777 177776 163176
 480 017054 012777 177777 163172
 481 017062 005077 163170

MOV #CRDY,GDDAT ;CONTROLLER READY
 BIT #DERR,@RLCS ;IF DRIVE ERROR SET
 BEQ 99\$;THE EXPECT IT LATER
 BIS #ERR!DERR,GDDAT
 99\$: MOV GDDAT,@RLCS ;LOAD CS
 MOV #-2,@RLBA ;LOAD BA WITH ALL 1'S
 MOV #-1,@RLDA ;LOAD RLDA
 CLR @RLMP ;WRITE RLMP

482
 483
 484
 485 017066 017737 163156 002364
 486 017074 042737 000001 002364
 487 017102 023737 002362 002364
 488 017110 001404

;CHECK IF RLCS IS OKAY
 MOV @RLCS,BDDAT ;READ RLCS
 BIC #DRDY,BDDAT ;IGNORE DRIVE READY
 CMP GDDAT,BDDAT ;RLCS OKAY?
 BEQ 1\$;YES, THEN BRANCH

489
 490 017112
 017112 104455
 017114 000311
 017116 006204
 017120 010210
 491 017122
 017122 104406

ERRDF 201,EM44,ERR2 ;MP MODIFIED CS
 TRAP CSERDF
 .WORD 201
 .WORD EM44
 .WORD ERR2
 1\$: CKLOOP ;CHECK IF /FL:LOE IS SET
 TRAP C\$CLP1

492
 493 017124 022777 177776 163120
 494 017132 001412
 495
 496 017134 012737 177776 002362
 497 017142 017737 163104 002364
 498

CMP #-2,@RLBA ;IS RLBA OKAY?
 BEQ 2\$;BRANCH IF OKAY
 MOV #-2,GDDAT ;SET UP EXPECTED
 MOV @RLBA,BDDAT ;READ RLBA

499 017150
 017150 104455
 017152 000323
 017154 006237
 017156 010210
 500 017160
 017160 104406

ERRDF 211,EM45,ERR2 ;MP MODIFIED BA
 TRAP CSERDF
 .WORD 211
 .WORD EM45
 .WORD ERR2
 2\$: CKLOOP ;CHECK IF /FL:LOE IS SET
 TRAP C\$CLP1

501 017162 022777 177777 163064
 502 017170 001412
 503
 504 017172 017737 163056 002364
 505 017200 012737 177777 002362
 506

CMP #-1,@RLDA ;DISK ADDRESS OKAY
 BEQ 3\$;YES, CONTINUE
 MOV @RLDA,BDDAT ;SET UP BAD
 MOV #-1,GDDAT ;SET UP EXPECTED

507 017206
 017206 104455
 017210 000324

ERRDF 212,EM46,ERR2 ;MP MODIFIED DA
 TRAP CSERDF
 .WORD 212


```

017212 006272          .WORD  EM46
017214 010210          .WORD  ERR2
508
509 017216          3$:
510
511
512 017216          ENDTST          ;****END OF TEST****
    017216          L10044:
    017216 104401      TRAP    C$ETST
513
514          .SBTTL  **TEST 21** - NOOP FUNCTION(RL11 ONLY)
515
516 017220          BGNTST          ;****START OF TEST****
517
518
519
520 017220          STARS
    :*****
    :TEST THAT NOOP WILL FUNCTION. WE WILL ISSUE THE
    :NOOP AND WAIT FOR CONTROLLER READY TO SET. A
    :TIMEOUT OF 200 MILLISECS IS ALLOWED. DRIVE 0 IS ALWAYS
    :SELECTED SINCE THE DRIVE IS NOT NECESSARY.
    STARS
    :*****
521
522
523
524
525 017220
526
527
528 017220 022737 000001 002410      CMP    #1,T.CNTRL      ;RLV11, OR RLV12?
529 017226 001010                      BNE    99$             ;YES SKIP TEST
530
531
532 017230 004537 013420          JSR    R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
533 017234 000000                      NOOP0
534 017236 004537 014306          JSR    R5,WTCRDY     ;NOOP(0) FUNCTION
535 017242          2$:          CKLOOP      ;WAIT FOR CONTROLLER READY HIGH
    017242 104406          TRAP    C$CLP1    ;CHECK IF /FL:LOE IS SET
536
537 017244 004537 013120          JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
538
539 017250          99$:
540 017250          ENDTST          ;****END OF TEST****
    017250          L10045:
    017250 104401      TRAP    C$ETST
541
542
543          .SBTTL  **TEST 22** - TEST NOOP DOES NOTHING (RL11 ONLY)
544
545 017252          BGNTST          ;****START OF TEST****
546
547 017252          STARS
    :*****
    :TEST THAT ISSUING A NOOP FUNCTION DOES NOTHING. THIS IS DONE BY WRITING
    :THE RLBA, AND RLDA, READING THE RLMP AND MAKING SURE NOTHING CHANGES.
    STARS
    :*****
548
549
550 017252
551
552 017252 022737 000001 002410      CMP    #1,T.CNTRL      ;RLV11, OR RLV12?
553 017260 001076                      BNE    3$             ;YES SKIP TEST.
  
```


594 017456 ENDTST :****END OF TEST****
 017456 L10046:
 017456 104401 TRAP C\$ETST

595
596
597 .SBTTL **TEST 23** - TEST OF INTERRUPT (RL11 ONLY)

598
599 017460 BGNTST :****START OF TEST****

600
601 017460

602 STARS
603 :*****
604 :CHECK THE INTERRUPT WITH A NOOP. WE WILL SET UP THE
605 :INTERRUPT VECTOR, LOWER THE PSW TO ZERO AND ISSUE
606 :A NOOP. THE INTERRUPT SERVICE ROUTINE WILL SET A
607 :FLAG UPON INTERRUPT AND RETURN IN LINE. WE WAIT 200 MILLISECONDS
608 :LOOKING FOR THAT FLAG TO BE SET BEFORE CALLING IT
609 :AN ERROR. IF THE INTERRUPT SENDS US TO ANOTHER
610 :VECTOR ADDRESS THEN THE ERROR HANDLER WILL REPORT
611 017460 :"TRAP TO XXXX FROM YYYY" AND RETURN TO DIAG SUP MONITOR. IF THE
 :INTERRUPT GOES TO ABOVE 1000 WHO KNOWS WHAT WILL HAPPEN.
 STARS
 :*****

612
613
614 017460 022737 000001 002410 CMP #1,T.CNTRL :RLV11 OR RLV12?
615 017466 001026 BNE 99\$:YES SKIP TEST.
616
617 017470 005037 002330 CLR INTFLG :CLEAR INTERRUPT OCCURRENCE FLAG
618 017474 SETPRI #PRI00 :SET PSW TO 0
 017474 012700 000000 MOV #PRI00,R0
619 017502 004537 013420 TRAP C\$SPRI
620 017506 000100 JSR R5,LDFUNC :ISSUE FUNCTION OF FOLLOWING WORD
621 017510 004537 014306 NOOP0!INTEN :NOOP AND INTERRUPT ENABLE
622 017514 005737 002330 JSR R5,WTCRDY :WAIT FOR CONTROLLER READY HIGH
623 017520 001004 TST INTFLG :DID INTERRUPT OCCUR
624 017522 BNE 2\$:IF SO BRANCH
 017522 104455 ERRDF 22,EM13,ERRO
 017524 000026 TRAP C\$ERDF
 017526 005252 .WORD 22
 017530 010160 .WORD EM13
 017532 005037 002330 .WORD ERRO
625 017532 005037 002330 2\$: CLR INTFLG
626 017536 CKLOOP :CHECK IF /FL:LOE IS SET
 017536 104406 TRAP C\$CLP1
627 017540 004537 013120 JSR R5,CHERR :CHECK CONTROLLER FOR ERRORS

628
629
630 017544 99\$:
631 017544 ENDTST :****END OF TEST****
 017544 L10047:
 017544 104401 TRAP C\$ETST

632
633 .SBTTL **TEST 24** - TEST PRIORITY BR LEVEL

634
635
636 017546 BGNTST :****START OF TEST****

637

638 017546

639
 640
 641
 642 017546

STARS
 :*****
 :TEST THAT PRIORITY GIVEN IS ACTUAL PRIORITY OF CONTROLLER. WE KNOW
 :THE BOARD WILL INTERRUPT. WE WILL START TRYING TO INTERRUPT AT 7
 :AND WORK DOWN TIL IT DOES INTERRUPT.
 STARS
 :*****

643
 644 017546 022737 000001 002410
 645 017554 002456
 646
 647 017556 012737 000340 002364
 648 017564 013737 002264 002362
 649

CMP #1,T.CNTRL ;RLV11??
 BLT 6\$;YES, SKIP TEST
 MOV #340,BDDAT ;SET UP INITIAL OF 7
 MOV BPRIOR,GDDAT ;GET GIVEN PRIORITY

650 017572
 017572 104404

BGNSEG ;****START OF SEGMENT****
 TRAP C\$BSEG

651
 652 017574 005037 002330
 653 017600
 017600 013700 002364
 017604 104441

5\$: CLR INTFLG ;CLEAR INTERRUPT OCCURRENCE
 SETPRI BDDAT ;SET PRIORITY
 MOV BDDAT,R0
 TRAP C\$SPRI

654
 655 017606 004537 013420
 656 017612 000100

JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
 NOOPO!INTEN

657
 658 017614 004537 014306
 659 017620
 017620 104410
 017622 000070

JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 TRAP C\$ESCAPE
 .WORD L10050-

660
 661 017624 004537 013120
 662 017630
 017630 104410
 017632 000060

JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
 TRAP C\$ESCAPE
 .WORD L10050-

663
 664 017634 023737 002364 002362
 665 017642 002012
 666
 667 017644 005737 002330
 668 017650 001004
 669

CMP BDDAT,GDDAT ;SHOULD IT INTERRUPT
 BGE 1\$;NO, BRANCH
 TST INTFLG ;DID INTERRUPT OCCUR
 BNE 2\$;YES, OK

670 017652
 017652 104455
 017654 000314
 017656 005406
 017660 010446

3\$: ERRDF 204,EM17,ERR7
 TRAP C\$ERDF
 .WORD 204
 .WORD EM17
 .WORD ERR7

671
 672 017662
 017662 104410
 017664 000014
 017666 000405

2\$: ESCAPE SEG ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
 TRAP C\$ESCAPE
 .WORD 10000\$-

673 017666 000405
 674 017670 005737 002330
 675 017674 001772
 676 017676 000765
 677

1\$: TST INTFLG ;DID INTERRUPT OCCUR
 BEQ 2\$;NO, OK
 BR 3\$;YES, ERROR

678 017700
 017700

ENDSEG ;****END OF SEGMENT****
 10000\$:


```

679 017700 104405          TRAP  C$ESEG
680 017702 162737 000040 002364 4$:  SUB  #40,BDDAT      ;NEXT LEVEL
681 017710 100331          BPL  5$
682 017712          6$:
683 017712          ENDTST
        017712          L10050:          ;*****END OF TEST*****
        017712 104401          TRAP  C$ETST
684          .SBTTL  **TEST 25** - GET STATUS FUNCTION
685          BGNTST          ;*****START OF TEST*****
686
687 017714
688
689
690 017714          STARS
        :*****
        :TEST GET STATUS FUNCTION. THE GET STATUS FUNCTION WILL
        :WORK IF DRIVE IS LOADED AND READY OR NOT. THE RLDA
        :IS LOADED WITH THE GET STATUS AND MARKER BITS (BITS 1,0)
        :AND THE FUNCTION IS ISSUED. WE WAIT 200 MILLISECONDS
        :FOR CONTROLLER READY. VERIFY THAT NO ERRORS OCCUR.
        :*****
        :*****
691
692
693
694
695
696 017714
697
698
699 017714 012777 000013 162332      MOV  #GSBIT!MK!DRST,@RLDA ;SET GET STATUS AND MARKER BIT
700 017722 004537 013420          JSR  R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
701 017726 000004          GSTAT          ;GET STATUS
702 017730 004537 014306          JSR  R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
703 017734          2$:  CKLOOP          ;CHECK IF /FL:LOE IS SET
        017734 104406          TRAP  C$CLP1
704
705 017736 004537 013120          JSR  R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
706
707 017742          ENDTST
        017742          L10051:          ;*****END OF TEST*****
        017742 104401          TRAP  C$ETST
708
709
710          .SBTTL  **TEST 26** - GET STATUS FUNCTION INTERRUPT
711          BGNTST          ;*****START OF TEST*****
712 017744
713          ;CHECK GET STATUS UNDER INTERRUPT
714
715
716 017744 005037 002330          CLR  INTFLG          ;CLEAR INTERRUPT OCCURANCE
717 017750          SETPRI #PRI00          ;PSW TO LEVEL 0
        017750 012700 000000          MOV  #PRI00,R0
        017754 104441          TRAP  C$SPRI
718 017756 012777 000003 162270      MOV  #GSBIT!MK,@RLDA ;SET UP DA
719 017764 004537 013420          JSR  R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
720 017770 000104          GSTAT!INTEN          ;GET STATUS, INT ENABLE
721 017772 004537 014306          JSR  R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
722 017776          SETPRI #PRI07
        017776 012700 000340          MOV  #PRI07,R0
        020002 104441          TRAP  C$SPRI
723 020004 005737 002330          TST  INTFLG          ;DID INTERRUPT OCCUR
  
```

```

724 020010 001004      BNE      2$          ;YES-BRANCH
725 020012      ERRDF   28.,EM30,ERRO
      020012 104455      TRAP   C$ERDF
      020014 000034      .WORD  28
      020016 005441      .WORD  EM30
      020020 010160      .WORD  ERRO
726 020022      2$:   CKLOOP          ;CHECK IF /FL:LOE IS SET
      020022 104406      TRAP   C$CLP1
727 020024 004537 013120 JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
728 020030 005037 002330 CLR    INTFLG        ;CLEAR INTERRUPT OCCURANCE
729 020034      SETPRI  #PRI00      ;PSW TO LEVEL 0
      020034 012700 000000 MOV    #PRI00,R0
      020040 104441      TRAP   C$SPRI
730 020042 012777 000003 162204 MOV    #GSBIT!MK,@RLDA ;SET UP DA FOR GET STATUS CMD
731 020050 004537 013420 JSR    R5,LDFUNC     ;ISSUE FUNCTION OF FOLLOWING WORD
732 020054 000004      GSTAT
733 020056 004537 014306 JSR    R5,WTCRDY     ;GET STATUS - SHOULD NOT CAUSE AN INTERRUPT
734 020062      SETPRI  #PRI07      ;WAIT FOR CONTROLLER READY HIGH
      020062 012700 000340 MOV    #PRI07,R0
      020066 104441      TRAP   C$SPRI
735 020070 005737 002330 TST   INTFLG        ;DID INTERRUPT OCCUR (SHOULD NOT)
736 020074 001404      BEQ    3$          ;NO - BRANCH (OK)
737 020076      ERRDF   281.,EM30A,ERRO
      020076 104455      TRAP   C$ERDF
      020100 000431      .WORD  281
      020102 005500      .WORD  EM30A
      020104 010160      .WORD  ERRO
738 020106      3$:   CKLOOP          ;CHECK IF /FL:LOE IS SET
      020106 104406      TRAP   C$CLP1
739 020110 004537 013120 JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
740 020114      ENDTST
      020114 104401      L10052: TRAP   C$SETST
      020114 104401      TRAP   C$SETST
741
742
743      .SBTTL **TEST 27** - GET STATUS FUNCTION GENERATES OPI W/O GS BIT
744
745 020116      BGNTST          ;****START OF TEST****
746
747 020116      STARS
      :*****
      :VERIFY THAT GET STATUS FUNCTION WILL NOT COMPLETE
      :WITHOUT SENDING OUT THE GET STATUS BIT IN THE RLDA.
      :WE SET MARKER BUT NO GET STATUS BIT IN THE RLDA AND
      :ISSUE A GET STATUS WE SHOULD RECIEVE AN OPI ERROR.
      :VERIFY THAT CONTROLLER READY SETS AND OPI SETS
      :*****
748
749
750
751
752
753 020116      STARS
      :*****
754
755
756 020116 012777 000001 162130 MOV    #MK,@RLDA     ;SET ONLY MARKER BIT!!
757 020124 004537 013420 JSR    R5,LDFUNC     ;ISSUE FUNCTION OF FOLLOWING WORD
758 020130 000004      GSTAT
759 020132 004537 014306 JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY HIGH
760 020136 032737 074000 002306 BIT    #74000,E.CS
761 020144 001405      BEQ    1$
762 020146 012737 004053 013402 MOV    #OPIERR,RESTMS
  
```



```

763 020154 004537 013120          JSR    R5,CHERR
764 020160          1$:    CKLOOP
      020160 104406          TRAP   C$CLP1
765 020162 032737 002000 002306    BIT    #OPI,E.CS          ;IS OPI SET?
766 020170 001004          BNE    2$                ;YES-BRANCH NO-CHECK TIMEOUT
767 020172          ERRDF  29.,EM33,ERRO
      020172 104455          TRAP   C$ERDF
      020174 000035          .WORD  29
      020176 005574          .WORD  EM33
      020200 010160          .WORD  ERRO
768 020202          2$:
769
770 020202          ENDTST                      ;****END OF TEST****
      020202 L10053:
      020202 104401          TRAP   C$SETST

771
772
773          .SBTTL  **TEST 28** - OPI UNDER INTERRUPT
774
775 020204          BGNTST                      ;****START OF TEST****
776 020204          STARS
      :*****
      :FORCE AN OPI ERROR UNDER INTERRUPT TO VERIFY THAT
      :AN INTERRUPT WILL OCCUR FROM OPI. THE OPI IS FORCED
      :USING A GET STATUS WITHOUT THE GET STATUS BIT SET
      :IN RLDA.
      :STARS
      :*****

782
783
784 020204          SETPRI  #PRI00
      020204 012700 000000    MOV    #PRI00,RO
      020210 104441          TRAP   C$SPRI
785 020212 005037 002330          CLR    INTFLG
786 020216 012777 000001 162030    MOV    #MK,@RLDA          ;SET ONLY MARKER BIT!!
787 020224 004537 013420          JSR    R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
788 020230 000104          GSTAT!INTEN              ;GET STATUS
789 020232 004537 014306          JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
790 020236          SETPRI  #PRI07
      020236 012700 000340    MOV    #PRI07,RO
      020242 104441          TRAP   C$SPRI
791 020244 005737 002330          TST    INTFLG              ;INTERRUPT OCCUR
792 020250 001004          BNE    2$
793 020252          ERRDF  30.,EM11,ERRO
      020252 104455          TRAP   C$ERDF
      020254 000036          .WORD  30
      020256 005211          .WORD  EM11
      020260 010160          .WORD  ERRO
794 020262          2$:    CKLOOP                      ;CHECK IF /FL:LOE IS SET
      020262 104406          TRAP   C$CLP1
795 020264 032737 074000 002306    BIT    #74000,E.CS
796 020272 001405          BEQ    1$
797 020274 012737 004053 013402    MOV    #OPIERR,RESTMS
798 020302 004537 013120          JSR    R5,CHERR
799 020306          1$:    CKLOOP
      020306 104406          TRAP   C$CLP1
800 020310 032737 002000 002306    BIT    #OPI,E.CS          ;IS OPI SET?
  
```

```

801 020316 001004          BNE      3$          :YES-BRANCH NO-CHECK TIMEOUT
802 020320          ERRDF   31.,EM33,ERRO
      020320 104455      TRAP   C$ERDF
      020322 000037      .WORD  31
      020324 005574      .WORD  EM33
      020326 010160      .WORD  ERRO
803 020330          3$:
804
805 020330          ENDTST          :****END OF TEST****
      020330          L10054:
      020330 104401      TRAP   C$SETST
806
807          .SBTTL  **TEST 29** - READ HEADER FUNCTION
808
809 020332          BGNTST          :****START OF TEST****
810 020332          STARS
      :*****
      :CHECK THAT READ HEADER WORKS, THAT WE CAN ISSUE
      :IT, GET READY BACK WITHOUT ANY ERRORS SETTING.
      STARS
      :*****
811
812
813 020332
814
815 020332 004537 013420      JSR    R5,LDFUNC      :ISSUE FUNCTION OF FOLLOWING WORD
816 020336 000010          RDHDR          :READ HEADER
817 020340 004537 014306      JSR    R5,WTCRDY      :WAIT FOR CONTROLLER READY HIGH READY
818 020344          2$:      CKLOOP          :CHECK IF /FL:LOE IS SET
      020344 104406      TRAP   C$CLP1
819 020346 004537 013120      JSR    R5,CHERR      :CHECK CONTROLLER FOR ERRORS
820
821 020352          ENDTST          :****END OF TEST****
      020352          L10055:
      020352 104401      TRAP   C$SETST
822
823          .SBTTL  **TEST 30** - READ HEADER FUNCTION INTERRUPT
824
825 020354          BGNTST          :****START OF TEST****
826
827 020354          STARS
      :*****
      :CHECK THAT READ HEADER WILL GENERATE AN INTERRUPT
      :UPON COMPLETION WITHOUT ANY ERRORS SETTING
      STARS
      :*****
828
829
830 020354
831
832
833 020354          SETPRI  #PRI00          :PSW TO 0
      020354 012700 000000      MOV    #PRI00,RO
      020360 104441      TRAP   C$SPRI
834 020362 005037 002330      CLR    INTFLG          :CLEAR INTERRUPT OCCURENCE
835 020366 004537 013420      JSR    R5,LDFUNC      :ISSUE FUNCTION OF FOLLOWING WORD
836 020372 000110          RDHDR!INTEN          :READ HEADER, INTR. ENA
837 020374 004537 014306      JSR    R5,WTCRDY      :WAIT FOR CONTROLLER READY HIGH
838 020400          SETPRI  #PRI07
      020400 012700 000340      MOV    #PRI07,RO
      020404 104441      TRAP   C$SPRI
839 020406 005737 002330      TST    INTFLG          :INTERRUPT HAPPEN
840 020412 001004          BNE      2$          :YES-CONTINUE
  
```



```

841 020414          ERRDF  35.,EM37,ERRO
      020414 104455    TRAP   C$ERDF
      020416 000043    .WORD  35
      020420 005716    .WORD  EM37
      020422 010160    .WORD  ERRO
842 020424          2$:    CKLOOP
      020424 104406    TRAP   C$CLP1          ;CHECK IF /FL:LOE IS SET
843
844 020426 004537 013120    JSR   R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
845
846 020432          ENDTST
      020432          L10056:
      020432 104401    TRAP   C$ETST          ;****END OF TEST****
847
848
849          .SBTTL  **TEST 31** - REPEATED RD HDRS YIELD SAME CYL AND HD
850
851 020434          BGNTST          ;****START OF TEST****
852
853
854 020434          STARS
      ;:*****
      ;:CHECK THAT READ HEADERS WILL RELIABLY READ THE SAME
      ;:CYLINDER AND HEAD SELECT. WE WILL READ HEADERS VERIFYING
      ;:THAT WE ALWAYS READ THE SAME CYLINDER AND HEAD SELECT.
      ;:STARS
      ;:*****
855
856
857
858 020434
859
860
861 020434 012701 000144    MOV   #100.,R1          ;SET UP TO DO 100 RD HDR'S
862 020440 004537 013420    JSR   R5,LDFUNC        ;ISSUE FUNCTION OF FOLLOWING WORD
863 020444 000010          RDHDR          ;READ HEADER
864 020446 004537 014306    JSR   R5,WTCRDY        ;WAIT FOR CONTROLLER READY HIGH
865 020452          99$:    ESCAPE  TST          ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      020452 104410    TRAP   C$ESCAPE
      020454 000122    .WORD  L10057-.
866
867 020456 004537 013120    JSR   R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
868 020462          ESCAPE  TST          ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      020462 104410    TRAP   C$ESCAPE
      020464 000112    .WORD  L10057-.
869
870 020466 013737 002314 002362    MOV   E.MP,GDDAT      ;READ FIRST HEADER (ASSUME GOOD)
871 020474 043737 002334 002362    BIC   SECMSK,GDDAT    ;MASK AWAY SECTOR BITS
872 020502          BGNSEG          ;****START OF SEGMENT****
      020502 104404    TRAP   C$BSEG
873 020504          2$:
874 020504 004537 013420    JSR   R5,LDFUNC        ;ISSUE FUNCTION OF FOLLOWING WORD
875 020510 000010          RDHDR
876 020512 004537 014306    JSR   R5,WTCRDY        ;WAIT FOR CONTROLLER READY HIGH
877 020516          97$:    ESCAPE  SEG          ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
      020516 104410    TRAP   C$ESCAPE
      020520 000054    .WORD  10000$-.
878
879 020522 004537 013120    JSR   R5,CHERR        ;CHECK CONTROLLER FOR ERRORS
880 020526          ESCAPE  SEG          ;IF /FL:LOE SET LOOP, ELSE EXIT SEG
      020526 104410    TRAP   C$ESCAPE
  
```

```

020530 000044 .WORD 10000$-.
881
882 020532 013737 002314 002364 MOV E.MP,BDDAT :READ HEADER
883 020540 043737 002334 002364 BIC SECMSK,BDDAT :MASK AWAY SECTOR BITS
884 020546 023737 002362 002364 CMP GDDAT,BDDAT :IS HEADER CORRECT
885 020554 001404 BEQ 4$
886
887 020556 ERRDF 36.,EM41,ERR4
020556 104455 TRAP C$ERDF
020560 000044 .WORD 36
020562 005756 .WORD EM41
020564 010324 .WORD ERR4
888
889 020566 4$: CKLOOP :CONSTANT CYL & HS
020566 104406 TRAP C$CLP1 :CHECK IF /FL:LOE IS SET
890
891 020570 DEC R1 :PERFORM ALL READ HDR'S
892 020572 001344 BNE 2$ :IF NOT GO BACK AND DO ANOTHER
893 020574 ENDSEG :*****END OF SEGMENT****
020574 10000$: TRAP C$ESEG
894 020576 ENDTST :*****END OF TEST****
020576 L10057: TRAP C$ETST
020576 104401
895
896
897 .SBTTL **TEST 32** - CHECK OF HEADER CRC
898
899 020600 BGNTST :*****START OF TEST****
900
901 020600 STARS
:*****
:CHECK THAT WE CAN READ THE HDCRC AFTER A
:READ HEADER AND THAT IT IS THE CORRECT CRC
:FOR THE HEADER.
902 STARS
903 :*****
904
905 020600
906
907
908 020600 005037 020650 CLR 3$
909 020604 004537 013420 JSR R5,LDFUNC :ISSUE FUNCTION OF FOLLOWING WORD
910 020610 000010 RDHDR :READ HEADER
911 020612 004537 014306 JSR R5,WTCRDY :WAIT FOR CONTROLLER READY HIGH
912 020616 ESCAPE TST :IF /FL:LOE SET LOOP, ELSE EXIT TST
020616 104410 TRAP C$ESCAPE
020620 000114 .WORD L10060-.
913
914 020622 004537 013120 JSR R5,CHERR :CHECK CONTROLLER FOR ERRORS
915 020626 ESCAPE TST :IF /FL:LOE SET LOOP, ELSE EXIT TST
020626 104410 TRAP C$ESCAPE
020630 000104 .WORD L10060-.
916
917 020632 013737 002314 020646 MOV E.MP,2$ :READ HEADER WORD CONTAINS SEC, HD, CYL
918
919 020640 004537 014032 JSR R5,SIMBCC :GO CALCULATE CRC
920 020644 000020 16. :16 BITS
921 020646 000000 2$: .WORD 0 :HEADER GOES HERE
  
```



```

922 020650 000000 3$: .WORD 0 ;START WITH 0 CRC
923 020652 013737 002344 020676 MOV CALBCC,5$ ;GET SECOND WORD IN SILO, CONTAINS 0'S
924 020660 013737 002316 020674 MOV E.MP1,4$
925 020666 004537 014032 JSR R5,SIMBCC
926 020672 000020 16.
927 020674 000000 4$: .WORD 0
928 020676 000000 5$: .WORD 0
929 020700 013737 002344 002362 MOV CALBCC,GDDAT ;STORE CALCULATED CRC AS GOOD
930 020706 013737 002320 002364 MOV E.MP2,BDDAT ;THIRD READ OF MP SILO GETS CRC
931 020714 023737 002362 002364 CMP GDDAT,BDDAT ;IS CRC CORRECT?
932 020722 001404 BEQ 6$ ;IF SO CONTINUE
933
934 020724 ERRDF 37.,EM42,ERR4
020724 104455 TRAP C$ERDF
020726 000045 .WORD 37
020730 006047 .WORD EM42
020732 010324 .WORD ERR4
935 020734 6$:
936
937 020734 ENDTST ;*****END OF TEST*****
020734 L10060:
020734 104401 TRAP C$ETST
938
939
940 .SBTTL **TEST 33** - CHECK CONSECUTIVE HEADERS
941
942 020736 BGNTST ;*****START OF TEST*****
943
944
945 020736 STARS
;*****
;CHECK THAT THE HEADERS ARE CONSECUTIVE. WE WILL DO
;40 (FORTY) READ HEADERS AND STORE EACH. AFTER WE HAVE
;READ THE FORTIETH HEADER WE WILL VERIFY THAT
;THEY CAME IN SEQUENTIAL, THAT 0 FOLLOWS 39,
;THAT THERE WERE NO ERRORS.
946 STARS
947 ;*****
948
949
950
951 020736
952
953
954 020736 005037 002366 CLR FIRST ;CLEAR FIRST READ DONE FLAG
955 020742 012703 003274 MOV #HDRBUF,R3 ;STORE HEADERS
956 020746 012701 000050 MOV #40.,R1 ;FORTY HEADERS
957 020752 012737 000210 002272 MOV #RDHDR!CRDY,B.CS
958 020760 053737 002270 002272 BIS DRIVE,B.CS
959 020766 013777 002272 161254 MOV B.CS,@RLCS
960 020774 042777 000200 161246 2$: BIC #200,@RLCS
961 021002 032777 000200 161240 1$: BIT #200,@RLCS ;DONE?
962 021010 001774 BEQ 1$
963 021012 017723 161232 MOV @RLCS,(R3)+
964 021016 017723 161234 MOV @RLMP,(R3)+
965 021022 017723 161230 MOV @RLMP,(R3)+
966 021026 017723 161224 MOV @RLMP,(R3)+
967 021032 005301 DEC R1 ;HAVE WE READ FORTY HEADERS
968 021034 001357 BNE 2$ ;GO BACK UNTIL FORTY DONE
969 021036 012703 003274 MOV #HDRBUF,R3 ;GET LIST OF HEADERS
970 021042 012701 000050 MOV #40.,R1 ;CHECK FORTY OF THEM
  
```

```

971 021046 011337 002306      MOV      (R3),E.CS
972 021052 005737 002306      TST      E.CS
973 021056 100016      BPL      99$
974 021060 012737 004312 013402      MOV      #RHDMS,RESTMS
975 021066 005723      TST      (R3)+
976 021070 012337 002314      MOV      (R3)+,E.MP
977 021074 012337 002316      MOV      (R3)+,E.MP1
978 021100 012337 002320      MOV      (R3)+,E.MP2
979 021104 004537 013120      JSR      R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
980 021110 000137 021252      JMP      7$
981 021114 005723      99$: TST      (R3)+
982 021116 011337 002364      MOV      (R3),BDDAT      ;GET HEADER
983 021122 005737 002366      TST      FIRST      ;IS THIS FIRST READ?
984 021126 001007      BNE      4$      ;NO, BRANCH
985 021130 012737 000001 002366      MOV      #1,FIRST      ;SET FIRST READ DONE FLAG
986 021136 013737 002364 002362 3$: MOV      BDDAT,GDDAT      ;SET UP NEXT READ EXPECTED
987 021144 000435      BR       6$      ;GO SEE IF TEST IS DONE
988 021146 005237 002362 4$: INC      GDDAT      ;INCREMENT EXP'D HEADER
989 021152 023737 002364 002362      CMP      BDDAT,GDDAT      ;IS NEW HEADER SEQUENTIAL?
990 021160 001766      BEQ      3$      ;YES THEN BRANCH
991 021162 033737 002334 002364      BIT      SECMSK,BDDAT      ;IS NEW HEADER ZERO?
992 021170 001015      BNE      5$      ;NO, THEN ERROR GO REPORT IT
993 021172 013737 002362 002346      MOV      GDDAT,TEMP2      ;YES, CHECK IF LAST HEADER WAS
994 021200 043737 002370 002346      BIC      CYLSK,TEMP2      ;MAX ADDRESS, IF SO BRANCH
995 021206 023737 002372 002346      CMP      MXSEC1,TEMP2      ;STORE NEW DATA AS OLD
996 021214 001750      BEQ      3$      ;AND PERFORM NEW RD HDR
997 021216 043737 002334 002362      BIC      SECMSK,GDDAT      ;EXPECTING ZERO SECTOR
998
999 021224      5$:
1000
1001 021224 005037 002366      CLR      FIRST      ;ERROR WILL MAKE US MISS
1002                                     ;NEXT SECTOR SEQUENTIALLY
1003                                     ;START OVER; CLEAR FIRST FLAG
1004 021230      ERRDF  38.,EM43,ERR2
1004 021230 104455      TRAP    C$ERRDF
1004 021232 000046      .WORD  38
1004 021234 006105      .WORD  EM43
1004 021236 010210      .WORD  ERR2
1005 021240      6$: CKLOOP
1005 021240 104406      TRAP    C$CLP1      ;CHECK IF /FL:LOE IS SET
1006
1007 021242 062703 000006      ADD      #6,R3
1008 021246 005301      DEC      R1      ;HAVE WE DONE THIS ENOUGH
1009 021250 001321      BNE      99$      ;NO, GO BACK DO IT AGAIN
1010 021252      7$:
1011 021252      ENDTST
1011 021252      L10061:
1011 021252 104401      TRAP    C$ETST
1012
1013
1014      .SBTTL  **TEST 34** - SEEK FUNCTION
1015
1016 021254      BGNST
1017 021254      STARS      ;****START OF TEST****
1018                                     ;*****
1019                                     ;CHECK THE SEEK FUNCTION RETURNS CONTROLLER READY
1019                                     ;WITH NO ERRORS. WE ISSUE A ONE TRACK IN WORD SEEK.

```



```
1020                                     :WE DO NOT CHECK THE RESULT FOR POSITION
1021 021254                               STARS
                                           :*****
1022
1023
1024 021254 012777 000205 160772          MOV    #BIT7!MK!SIGN,@RLDA  ;SET UP DA-DIFF=1,MARKER,TOWARDS
1025 021262 004537 013420                JSR    R5,LDFUNC           ;ISSUE FUNCTION OF FOLLOWING WORD
1026 021266 000006                        SEEK                               ;SEEK
1027 021270 004537 014306                JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
1028 021274 012737 000010 002414          MOV    #8.,DLYCNT        ;INITIALIZE DELAY COUNT
1029 021302                                WAIT1: DELAY 250.           ;IMPLEMENT TIME DELAY
                                           MOV    #250.,(PC)+
                                           .WORD 0
                                           MOV    LSDLY,(PC)+
                                           .WORD 0
                                           DEC    -6(PC)
                                           BNE   -4
                                           DEC    -22(PC)
                                           BNE   -20
1030 021332 005337 002414                DEC    DLYCNT             ;DECREMENT DELAY COUNT
1031 021336 001361                        BNE   WAIT1              ;BRANCH IF DELAY NOT EXPIRED
1032 021340                                2$:  CKLOOP              ;CHECK IF /FL:LOE IS SET
                                           TRAP  C$CLP1
1033 021342 004537 013120                JSR    R5,CHERR           ;CHECK CONTROLLER FOR ERRORS
1034
1035 021346                                ENDTST
                                           L10062: TRAP  C$SETST    ;****END OF TEST****
                                           021346 104401
1036
1037
1038                                     .SBTTL **TEST 35** - CHECK DRIVE READY ON SEEK
1039
1040 021350                                BGNTST                      ;****START OF TEST****
1041
1042
1043 021350                               STARS
                                           :*****
                                           :CHECK THE SEEK FUNCTION RETURNS DRIVE READY WITH
                                           :NO ERRORS. WE ISSUE A ONE TRACK INWARD SEEK. WE DO
                                           :NOT CHECK THE RESULT FOR POSITION
1044                                     STARS
                                           :*****
1045
1046
1047 021350                               STARS
                                           :*****
1048
1049
1050
1051 021350 012777 000201 160676          MOV    #BIT7!MK,@RLDA  ;SET DA, MARKER, DIFF=1.
1052 021356 004537 013420                JSR    R5,LDFUNC           ;ISSUE FUNCTION OF FOLLOWING WORD
1053 021362 000006                        SEEK                               ;SEEK
1054 021364 004537 014306                JSR    R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
1055 021370                                CKLOOP              ;CHECK IF /FL:LOE IS SET
                                           TRAP  C$CLP1
1056
1057 021372 004537 013120                JSR    R5,CHERR           ;CHECK CONTROLLER FOR ERRORS
1058 021376                                CKLOOP              ;CHECK IF /FL:LOE IS SET
                                           TRAP  C$CLP1
                                           021376 104406
1059
1060 021400 004537 014220                JSR    R5,WTCRDY          ;WAIT FOR DRIVE READY
```

```

1061 021404          CKLOOP          ;CHECK IF /FL:LOE IS SET
      021404 104406 TRAP          C$CLP1
1062
1063 021406 004537 013120 JSR          R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
1064
1065 021412          ENDTST          ;****END OF TEST****
      021412          L10063:
      021412 104401 TRAP          C$ETST
1066
1067
1068          .SBTTL  **TEST 36** - SEEK FUNCTION INTERRUPT
1069
1070 021414          BGNTST          ;****START OF TEST****
1071
1072
1073 021414          STARS
      :*****
      :CHECK THAT CONTROLLER READY RESETTING WHEN THE SEEK IS
      :INITIATED CAUSES AN INTERRUPT BUT DRIVE READY WILL
      :NOT. WE ALSO MONITOR FOR ANY ERROR BITS SETTING.
      :*****
1074
1075
1076
1077 021414          STARS
      :*****
1078
1079
1080
1081
1082 021414 005037 002330 CLR          INTFLG
1083 021420          SETPRI          #PRI00          ;SET PSW TO 0
      021420 012700 000000 MOV          #PRI00,R0
      021424 104441 TRAP          C$SPRI
1084 021426 012777 000205 160620 MOV          #BIT7!MK!SIGN,@RLDA ;SET UP RLDA
1085 021434 004537 013420 JSR          R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
1086 021440 000106 SEEK!INTEN ;SEEK AND INTR. ENA.
1087 021442 004537 014306 JSR          R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
1088 021446 000240 NOP
1089 021450 005737 002330 1$: TST          INTFLG          ;DID INTERRUPT OCCUR
1090 021454 001004 BNE          2$          ;YES, GO CHECK DRDY
1091 021456          ERRDF          40,EM47,ERRO
      021456 104455 TRAP          C$ERDF
      021460 000050 .WORD          40
      021462 006325 .WORD          EM47
      021464 010160 .WORD          ERRO
1092 021466          2$: CKLOOP          ;CHECK IF /FL:LOE IS SET
      021466 104406 TRAP          C$CLP1
1093
1094
1095 021470 004537 013120 JSR          R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
1096 021474          CKLOOP          ;CHECK IF /FL:LOE IS SET
      021474 104406 TRAP          C$CLP1
1097
1098 021476 005037 002330 CLR          INTFLG          ;CLEAR INTERRUPT OCCURANCE
1099
1100
1101 021502 004537 014220 5$: JSR          R5,WTD RDY          ;WAIT FOR DRIVE READY
1102 021506          CKLOOP          ;CHECK IF /FL:LOE IS SET
      021506 104406 TRAP          C$CLP1
1103
  
```



```

1104 021510          SETPRI  #PRI07
      021510 012700 000340  MOV    #PRI07,R0
      021514 104441  TRAP   C$SPRI
1105 021516 005737 002330  TST   INTFLG      ;DID DRIVE READY CAUSE INTERRUPT
1106 021522 001404      BEQ    6$             ;NO, CONTINUE
1107
1108 021524          ERRDF  42.,EM52,ERRO
      021524 104455  TRAP   C$ERDF
      021526 000052  .WORD  42
      021530 006356  .WORD  EM52
      021532 010160  .WORD  ERRO
1109 021534          6$:   CKLOOP
      021534 104406  TRAP   C$CLP1      ;CHECK IF /FL:LOE IS SET
1110
1111 021536          ENDTST
      021536          L10064:
      021536 104401  TRAP   C$ETST      ;****END OF TEST****
1112
1113
1114          .SBTTL  **TEST 37** - TEST DIFFERENCE WORD TRANSMISSION
1115
1116 021540          BGNST          ;****START OF TEST****
1117
1118
1119
1120
1121 021540          STARS
      :*****
      :VERIFY THAT THE DIFFERENCE WORD LOADS AND IS
      :TRANSMITTED CORRECTLY. WE WILL ISSUE SEEKS WITH THE
      :DIFFERENCE WORD CONTAINING ALL OF THE BIT PATTERNS FLOATING 1,
      :GROWING 1, GROWING 0 AND SHITING 0. THE SEEK WILL
      :START FROM TRACK 0 EACH TIME AND WILL RETURN THERE
      :EACH, THUS BOTH DIRECTIONS FOR PATTERNS WILL BE CHECKED.
      :READ HEADERS ARE USED TO VERIFY THE SEEK CORRECTNESS.
      :ERRORS ARE MONITORED AND REPORTED.
      STARS
      :*****
1122
1123
1124
1125
1126
1127
1128
1129
1130 021540
1131
1132
1133 021540 012703 002626  BGNSEG  MOV    #SKLST,R3      ;GET LIST OF DIFFERENCE WORDS
1134 021544          TRAP   C$BSEG      ;****START OF SEGMENT****
      021544 104404
1135 021546          1$:   JSR    R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
1136 021546 004537 013420  RDHDR
1137 021552 000010          JSR    R5,WTCRDY      ;READ HEADER
1138 021554 004537 014306  98$:   CKLOOP      ;WAIT FOR CONTROLLER READY HIGH
1139 021560          TRAP   C$CLP1      ;CHECK IF /FL:LOE IS SET
      021560 104406
1140
1141 021562 004537 013120  JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1142 021566          TRAP   C$CLP1      ;CHECK IF /FL:LOE IS SET
      021566 104406
1143
1144 021570 013737 002314 002364  MOV    E.MP,BDDAT      ;READ HEADER
1145 021576 043737 002334 002364  BIC    SECMSK,BDDAT    ;CLEAR OUT SECTOR
1146 021604 001462          BEQ    99$             ;IF ON TRACK ZERO, H.S. ZERO, OK
  
```

```

1147
1148
1149
1150
1151 021606 042737 000100 002364      BIC      #RHHS,BDDAT      ;CLEAR OUT HEAD SELECT
1152 021614 013777 002364 160432      MOV      BDDAT,@RLDA  ;PUT CYLINDER AS DIFFERENCE WORD
1153 021622 052777 000001 160424      BIS      #MK,@RLDA   ;SET MARKER BIT
1154 021630 004537 013420      JSR      R5,LDFUNC    ;ISSUE FUNCTION OF FOLLOWING WORD
1155 021634 000006                      SEEK     ;SEEK
1156 021636 004537 014306      JSR      R5,WTCRDY   ;WAIT FOR CONTROLLER READY HIGH
1157 021642                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021642 104406      TRAP    C$CLP1

1158
1159 021644 004537 013120      JSR      R5,CHERR    ;CHECK CONTROLLER FOR ERRORS
1160 021650                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021650 104406      TRAP    C$CLP1

1161
1162 021652 004537 014220      JSR      R5,WTD RDY  ;WAIT FOR DRIVE READY
1163 021656                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021656 104406      TRAP    C$CLP1

1164
1165 021660 004537 013120      JSR      R5,CHERR    ;CHECK CONTROLLER FOR ERRORS
1166 021664                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021664 104406      TRAP    C$CLP1

1167
1168 021666 004537 013420      JSR      R5,LDFUNC    ;ISSUE FUNCTION OF FOLLOWING WORD
1169 021672 000010      RDHDR   ;READ HEADER
1170 021674 004537 014306      JSR      R5,WTCRDY   ;WAIT FOR CONTROLLER READY HIGH
1171 021700                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021700 104406      TRAP    C$CLP1

1172
1173 021702 004537 013120      JSR      R5,CHERR    ;CHECK CONTROLLER FOR ERRORS
1174 021706                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021706 104406      TRAP    C$CLP1

1175
1176 021710 005037 002362      CLR      GDDAT       ;CLEAR EXPECTED
1177 021714 013737 002364 002376      MOV      BDDAT,DWORD ;SAVE DIFFERENCE WORD
1178 021722 013737 002314 002364      MOV      E.MP,BDDAT  ;READ HEADER
1179 021730 043737 002334 002364      BIC      SECMSK,BDDAT ;MASK OUT SECTOR BITS
1180 021736 001404                      BEQ     5$           ;BRANCH IF ON ZERO TRACK
1181
1182 021740                      ERRDF   43.,EM54,ERR3
      021740 104455      TRAP    C$ERDF
      021742 000053      .WORD  43
      021744 006426      .WORD  EM54
      021746 010252      .WORD  ERR3
1183 021750                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021750 104406      TRAP    C$CLP1

1184
1185 021752 011377 160276                      MOV      (R3),@RLDA  ;GET DIFFERENCE WORD
1186 021756 052777 000005 160270      BIS      #SIGN!MK,@RLDA ;SET SIGN (TOWARDS SPINDLE) AND MARKER
1187 021764 004537 013420      JSR      R5,LDFUNC    ;ISSUE FUNCTION OF FOLLOWING WORD
1188 021770 000006                      SEEK     ;SEEK
1189 021772 004537 014306      JSR      R5,WTCRDY   ;WAIT FOR CONTROLLER READY HIGH
1190 021776                      CKLOOP  ;CHECK IF /FL:LOE IS SET
      021776 104406      TRAP    C$CLP1
1191
  
```


1192	022000	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1193	022004				CKLOOP		:CHECK IF /FL:LOE IS SET
	022004	104406			TRAP	C\$CLP1	
1194							
1195	022006	004537	014220		JSR	R5,WTDRDY	:WAIT FOR DRIVE READY
1196	022012			87\$:	CKLOOP		:CHECK IF /FL:LOE IS SET
	022012	104406			TRAP	C\$CLP1	
1197							
1198	022014	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1199	022020				CKLOOP		:CHECK IF /FL:LOE IS SET
	022020	104406			TRAP	C\$CLP1	
1200							
1201	022022	004537	013420		JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
1202	022026	000010			RDHDR		:READ HEADER
1203							
1204	022030	004537	014306		JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
1205	022034				CKLOOP		:CHECK IF /FL:LOE IS SET
	022034	104406			TRAP	C\$CLP1	
1206							
1207	022036	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1208	022042				ESCAPE	SEG	:IF /FL:LOE SET LOOP, ELSE EXIT SEG
	022042	104410			TRAP	C\$ESCAPE	
	022044	000106			.WORD	10000\$-	
1209							
1210	022046	011337	002362		MOV	(R3),GDDAT	:GET EXPECTED CYLINDER
1211	022052	011337	002376		MOV	(R3),DWORD	:SET UP DIFFERENCE FOR SEEK
1212	022056	013737	002314	002364	MOV	E,MP,BDDAT	:READ HEADER FROM RLMP
1213	022064	043737	002334	002364	BIC	SECMSK,BDDAT	:CLEAR OUT SECTOR BITS
1214	022072	023737	002362	002364	CMP	GDDAT,BDDAT	:DID SEEK GO TO THE RIGHT
1215	022100	001404			BEQ	9\$:TRACK, IF SO, GO GET NEXT
1216							
1217	022102				ERRDF	44,EM54,ERR3	
	022102	104455			TRAP	C\$ERDF	
	022104	000054			.WORD	44	
	022106	006426			.WORD	EM54	
	022110	010252			.WORD	ERR3	
1218	022112				9\$:	CKLOOP	:CHECK IF /FL:LOE IS SET
	022112	104406			TRAP	C\$CLP1	
1219							
1220	022114	005723			TST	(R3)+	:BUMP PATTERN
1221	022116	023727	002406	000001	CMP	T,DRIVE,#1	
1222	022124	001005			BNE	2\$	
1223	022126	020327	002726		CMP	R3,#SKEND	
1224	022132	001407			BEQ	10\$	
1225	022134	000137	021546		JMP	1\$	
1226							
1227	022140	020327	002770		2\$:	CMP	R3,#SKEEND
1228	022144	001402			BEQ	10\$	
1229	022146	000137	021546		JMP	1\$	
1230							
1231	022152				10\$:		
1232							
1233	022152				ENDSEG		:****END OF SEGMENT****
	022152				10000\$:		
	022152	104405			TRAP	C\$ESEG	
1234	022154				ENDTST		:****END OF TEST****
	022154				L10065:		

```

022154 104401 TRAP CSETST
1235
1236
1237 .SBTTL **TEST 38** - VERIFY HEAD SELECT 0 VIA RD HDR
1238
1239 022156 BGNTST ;****START OF TEST****
1240
1241 ;
1242
1243 022156 STARS
:*****
:CHECK THAT WE CAN SELECT HEAD SELECT ZERO. ISSUE
:SEEK TO HEAD SELECT 0 AND VERIFY WITH READ HEADER.
STARS
:*****
1244
1245
1246 022156
1247
1248 022156 012777 000001 160070 99$: MOV #MK,@RLDA ;SET MARKER IN RLDA
1249 022164 005037 002362 CLR GDDAT ;SET EXPECTED
1250 ;LOAD HS=0 INTO RLDA
1251 022170 2$: JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
1252 022170 004537 013420 SEEK ;SEEK
1253 022174 000006 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
1254 022176 004537 014306 CKLOOP ;CHECK IF /FL:LOE IS SET
1255 022202 TRAP C$CLP1
022202 104406
1256
1257 022204 004537 013120 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
1258 022210 CKLOOP ;CHECK IF /FL:LOE IS SET
022210 104406 TRAP C$CLP1
1259
1260 022212 004537 014220 89$: JSR R5,WTD RDY ;WAIT FOR DRIVE READY
1261 022216 CKLOOP ;CHECK IF /FL:LOE IS SET
022216 104406 TRAP C$CLP1
1262
1263 022220 004537 013120 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
1264 022224 CKLOOP ;CHECK IF /FL:LOE IS SET
022224 104406 TRAP C$CLP1
1265
1266 022226 004537 013420 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
1267 022232 000010 RDHDR ;READ HEADER
1268 022234 004537 014306 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
1269 022240 96$: CKLOOP ;CHECK IF /FL:LOE IS SET
022240 104406 TRAP C$CLP1
1270
1271 022242 004537 013120 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
1272 022246 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
022246 104410 TRAP C$ESCAPE
022250 000036 .WORD L10066-.
1273
1274 022252 013737 002314 002364 MOV E.MP,BDDAT ;READ HEADER FOR HEAD SELECT
1275 022260 042737 177677 002364 BIC #177677,BDDAT ;MASK ONLY HEAD SELECT
1276 022266 023737 002362 002364 CMP GDDAT,BDDAT ;COMPARE HEAD SELECTS
1277 022274 001404 BEQ 5$ ;IF EQUAL CONTINUE
1278
1279 022276 ERRDF 45,EM55,ERR4
022276 104455 TRAP C$ERDF
022300 000055 .WORD 45

```



```

022302 006465      .WORD  EM55
022304 010324      .WORD  ERR4
1280 022306      5$:
1281
1282 022306      ENDTST      ;****END OF TEST****
      022306      L10066:
      022306 104401      TRAP    CSETST

1283
1284
1285      .SBTTL  **TEST 39** - VERIFY HEAD SELECT 1 VIA RD HDR
1286
1287 022310      BGNTST      ;****START OF TEST****
1288
1289
1290 022310      STARS
      :*****
      :CHECK THAT WE CAN SELECT HEAD SELECT ONE.  ISSUE
      :SEEK TO HEAD SELECT 1 AND VERIFY WITH READ HEADER.
1291
1292
1293 022310      STARS
      :*****
  
```

```

1294
1295
1296 022310 012777 000001 157736 99$:  MOV    #MK,@RLDA      ;SET MARKER IN RLDA
1297 022316 052777 000020 157730 2$:  BIS    #DAHS,@RLDA    ;LOAD HS=1 INTO RLDA
1298 022324 004537 013420      JSR    R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
1299 022330 000006      SEEK
1300 022332 004537 014306      JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY HIGH
1301 022336      CKLOOP      ;CHECK IF /FL:LOE IS SET
      022336 104406      TRAP    C$CLP1

1302
1303 022340 004537 013120      JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1304 022344      CKLOOP      ;CHECK IF /FL:LOE IS SET
      022344 104406      TRAP    C$CLP1

1305
1306 022346 004537 014220      JSR    R5,WTD RDY    ;WAIT FOR DRIVE CLEAR
1307 022352      CKLOOP      ;CHECK IF /FL:LOE IS SET
      022352 104406      TRAP    C$CLP1

1308
1309 022354 004537 013120      JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1310 022360      CKLOOP      ;CHECK IF /FL:LOE IS SET
      022360 104406      TRAP    C$CLP1

1311
1312 022362 004537 013420      JSR    R5,LDFUNC      ;ISSUE FUNCTION OF FOLLOWING WORD
1313 022366 000010      RDHDR      ;READ HEADER
1314 022370 004537 014306      JSR    R5,WTCRDY     ;WAIT FOR CONTROLLER READY HIGH
1315 022374      CKLOOP      ;CHECK IF /FL:LOE IS SET
      022374 104406      TRAP    C$CLP1

1316
1317 022376 004537 013120      JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1318 022402      ESCAPE      ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      022402 104410      TRAP    C$ESCAPE
      022404 000044      .WORD  L10067-.

1319
1320 022406 013737 002314 002364      MOV    E.MP,BDDAT    ;READ HEADER
1321 022414 042737 177677 002364      BIC    #177677,BDDAT ;MASK FOR H.S.
1322 022422 012737 000100 002362      MOV    #RHHS,GDDAT   ;SET EXPECTED
1323 022430 023737 002362 002364      CMP    GDDAT,BDDAT   ;CORRECT HEAD
  
```

```

1324 022436 001404          BEQ      5$          :YES, CONTINUE
1325
1326 022440          ERRDF    46,EM55,ERR4
      022440 104455          TRAP    C$ERRDF
      022442 000056          .WORD   46
      022444 006465          .WORD   EM55
      022446 010324          .WORD   ERR4
1327 022450          5$:
1328
1329 022450          ENDTST          :****END OF TEST****
      022450          L10067:
      022450 104401          TRAP    C$TST

.SBTTL **TEST 40** - VERIFY HEAD SELECT 0 VIA GET STATUS
1330
1331
1332
1333
1334 022452          BGNTST          :****START OF TEST****
1335
1336 022452          STARS
      :*****
      :CHECK THAT WE CAN READ BACK HEAD SELECT 0 WITH
      :A GET STATUS FUNCTION.  SELECT H.S. 0 WITH A SEEK
      :VERIFY WITH GET STATUS
      STARS
      :*****
1337
1338
1339
1340 022452
1341
1342 022452 012777 000001 157574      MOV     #MK,@RLDA      :SET MARKER IN RLDA
1343
1344 022460 005037 002362          2$: CLR     GDDAT        :LOAD HS=0 INTO RLDA
1345 022464 004537 013420          3$: JSR    R5,LDFUNC     :SET UP EXP'D
1346 022470 000006          :ISSUE FUNCTION OF FOLLOWING WORD
1347 022472 004537 014306          JSR    R5,WTCRDY      :SEEK
1348 022476          CKLOOP           :WAIT FOR CONTROLLER READY HIGH
      022476 104406          TRAP   C$CLP1        :CHECK IF /FL:LOE IS SET
1349
1350 022500 004537 013120          JSR    R5,CHERR       :CHECK CONTROLLER FOR ERRORS
1351 022504          CKLOOP           :CHECK IF /FL:LOE IS SET
      022504 104406          TRAP   C$CLP1
1352
1353 022506 004537 014220          JSR    R5,WTD RDY     :WAIT FOR DRIVE READY
1354 022512          CKLOOP           :CHECK IF /FL:LOE IS SET
      022512 104406          TRAP   C$CLP1
1355
1356 022514 004537 013120          JSR    R5,CHERR       :CHECK CONTROLLER FOR ERRORS
1357 022520          CKLOOP           :CHECK IF /FL:LOE IS SET
      022520 104406          TRAP   C$CLP1
1358
1359 022522 012777 000003 157524      MOV     #GSBIT!MK,@RLDA :SET UP FOR GET STATUS IN DA
1360 022530 004537 013420          JSR    R5,LDFUNC     :ISSUE FUNCTION OF FOLLOWING WORD
1361 022534 000004          GSTAT           :GET STATUS
1362 022536 004537 014306          JSR    R5,WTCRDY      :WAIT FOR CONTROLLER READY HIGH
1363 022542          CKLOOP           :CHECK IF /FL:LOE IS SET
      022542 104406          TRAP   C$CLP1
1364
1365 022544 004537 013120          JSR    R5,CHERR       :CHECK CONTROLLER FOR ERRORS
1366 022550          ESCAPE          :IF /FL:LOE SET LOOP, ELSE EXIT TST
      022550 104410          TRAP   C$ESCAPE
  
```



```

022552 000036          .WORD  L10070-.
1367
1368 022554 013737 002314 002364  MOV  E.MP,BDDAT  ;READ STATUS FOR HEAD SELECT BIT
1369 022562 042737 177677 002364  BIC  #177677,BDDAT ;LEAVE ONLY H.S. BIT
1370 022570 023737 002362 002364  CMP  GDDAT,BDDAT  ;IS HEAD SELECT CORRECT?
1371 022576 001404          BEQ  6$           ;YES, CONTINUE
1372
1373 022600          ERRDF  47.,EM56,ERR4
      022600 104455  TRAP  C$ERDF
      022602 000057  .WORD  47
      022604 006520  .WORD  EM56
      022606 010324  .WORD  ERR4
1374 022610          6$:
1375
1376 022610          ENDTST          ;****END OF TEST****
      022610          L10070:
      022610 104401  TRAP  C$ETST

.SBTTL **TEST 41** - VERIFY HEAD SELECT 1 VIA GET STATUS
BGNTST          ;****START OF TEST****

STARS
:*****
:CHECK THAT WE CAN READ BACK HEAD SELECT 1 WITH A GET
:STATUS FUNCTION.  SELECT H.S. 1 WITH A SEEK AND VERIFY WITH
:GET STATUS
STARS
:*****

1384
1385
1386
1387 022612          2$:
      022612          3$:
1388
1389
1390 022612 012777 000001 157434  MOV  #MK,@RLDA  ;SET MARKER IN RLDA
1391 022620 052777 000020 157426  BIS  #DAHS,@RLDA ;LOAD HS=1 INTO RLDA
1392 022626 012737 000100 002362  MOV  #STHS,GDDAT ;SET UP EXP'D
1393 022634 004537 013420          JSR  R5,LDFUNC  ;ISSUE FUNCTION OF FOLLOWING WORD
1394 022640 000006          SEEK          ;SEEK
1395 022642 004537 014306          JSR  R5,WTCRDY  ;WAIT FOR CONTROLLER READY HIGH
1396 022646          CKLOOP
      022646 104406  TRAP  C$CLP1  ;CHECK IF /FL:LOE IS SET
1397
1398 022650 004537 013120          JSR  R5,CHERR  ;CHECK CONTROLLER FOR ERRORS
1399 022654          CKLOOP
      022654 104406  TRAP  C$CLP1  ;CHECK IF /FL:LOE IS SET
1400
1401 022656 004537 014220          JSR  R5,WTD RDY ;WAIT FOR DRIVE READY
1402 022662          CKLOOP
      022662 104406  TRAP  C$CLP1  ;CHECK IF /FL:LOE IS SET
1403
1404 022664 004537 013120          JSR  R5,CHERR  ;CHECK CONTROLLER FOR ERRORS
1405 022670          CKLOOP
      022670 104406  TRAP  C$CLP1  ;CHECK IF /FL:LOE IS SET
1406
1407 022672 012777 000003 157354  MOV  #GSBIT!MK,@RLDA ;SET UP FOR GET STATUS IN DA
1408 022700 004537 013420          JSR  R5,LDFUNC  ;ISSUE FUNCTION OF FOLLOWING WORD
1409 022704 000004          GSTAT          ;GET STATUS
1410 022706 004537 014306          JSR  R5,WTCRDY  ;WAIT FOR CONTROLLER READY HIGH
  
```

```

1411 022712          ESCAPE TST          ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      022712 104410  TRAP   C$ESCAPE
      022714 000046  .WORD  L10071-.
1412
1413 022716 004537 013120 JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1414 022722          ESCAPE TST          ;IF /FL:LOE SET LOOP, ELSE EXIT TST
      022722 104410  TRAP   C$ESCAPE
      022724 000036  .WORD  L10071-.
1415
1416 022726 013737 002314 002364 MOV   E,MP,BDDAT    ;READ STATUS FOR HEAD SELECT BIT
1417 022734 042737 177677 002364 BIC   #177677,BDDAT ;LEAVE ONLY H.S. BIT
1418 022742 023737 002362 002364 CMP   GDDAT,BDDAT  ;IS HEAD SELECT CORRECT?
1419 022750 001404          BEQ   6$             ;YES, CONTINUE
1420
1421 022752          ERRDF 48,EM56,ERR4
      022752 104455  TRAP   C$ERDF
      022754 000060  .WORD  48
      022756 006520  .WORD  EM56
      022760 010324  .WORD  ERR4
1422 022762          6$:
1423
1424 022762          ENDTST          ;****END OF TEST****
      022762          L10071:
      022762 104401  TRAP   C$ETST
1425
1426
1427          .SBTTL **TEST 42** - TEST TIME AT WHICH DIF WD GETS TRANSMITTED
1428
1429 022764          BGNTST          ;****START OF TEST****
1430
1431
1432 022764          STARS
      :*****
      :VERIFY THAT THE DIFFERENCE WORD ON A SEEK IS
      :TRANSMITTED PRIOR TO CONTROLLER READY SETTING. THIS
      :IS DONE BY SETTING A KNOWN DIFFERENCE WORD IN
      :THE RLDA ISSUING A A SEEK, WAITING FOR CONTROLLER READY
      : (BUT NOT DRIVE READY), WRITING A DIFFERENT RLDA AND WAITING
      :FOR DRIVE READY. THE RESULTANT POSITION SHOULD BE THAT
      :OF THE FIRST RLDA ONLY.
      :*****
1433
1434
1435
1436
1437
1438
1439
1440 022764          STARS
      :*****
1441
1442
1443 022764 004537 013420 JSR    R5,LDFUNC    ;ISSUE FUNCTION OF FOLLOWING WORD
1444 022770 000010          RDHDR          ;READ HEADER
1445 022772 004537 014306 JSR    R5,WTCRDY    ;WAIT FOR CONTROLLER READY HIGH
1446 022776          CKLOOP          ;CHECK IF /FL:LOE IS SET
      022776 104406  TRAP   C$CLP1
1447
1448 023000 004537 013120 JSR    R5,CHERR      ;CHECK CONTROLLER FOR ERRORS
1449 023004          CKLOOP          ;CHECK IF /FL:LOE IS SET
      023004 104406  TRAP   C$CLP1
1450
1451 023006 013737 002314 002362 MOV   E,MP,GDDAT    ;READ HEADER
1452 023014 043737 002334 002362 BIC   SECMSK,GDDAT ;CLEAR SECTOR BITS
1453 023022 012777 000001 157224 MOV   #MK,@RLDA    ;SET MARKER IN RLDA
  
```


1454	023030	032737	000100	002362	BIT	#RHHS,GDDAT	:TEST H.S.
1455	023036	001403			BEQ	2\$:IF ZERO, CONTINUE
1456	023040	052777	000020	157206	BIS	#DAHS,@RLDA	:ONE, SET SO WE WILL REMAIN THERE
1457	023046	013737	002362	002354	MOV	GDDAT,TMPO	:STORE HEADER
1458	023054	042737	000100	002354	BIC	#RHHS,TMPO	:CLEAR H.S. FROM STORED WORD
1459	023062	023727	002406	000001	CMP	T.DRIVE,#1	
1460	023070	001034			BNE	12\$	
1461	023072	023737	002354	002704	CMP	TMPO,HALMAX	
1462	023100	101007			BHI	3\$	
1463	023102	052777	000004	157144	BIS	#SIGN,@RLDA	
1464	023110	063737	002702	002362	ADD	QUAMAX,GDDAT	
1465	023116	000403			BR	4\$	
1466	023120	163737	002702	002362	SUB	QUAMAX,GDDAT	
1467	023126	053777	002702	157120	BIS	QUAMAX,@RLDA	
1468	023134	012737	000001	002356	MOV	#MK,TMP1	
1469	023142	032777	000020	157104	BIT	#DAHS,@RLDA	
1470	023150	001037			BNE	5\$	
1471	023152	052737	000020	002356	BIS	#DAHS,TMP1	
1472	023160	000433			BR	5\$	
1473	023162	023737	002354	002734	CMP	TMPO,HMAX	
1474	023170	101007			BHI	13\$	
1475	023172	052777	000004	157054	BIS	#SIGN,@RLDA	
1476	023200	063737	002732	002362	ADD	QMAX,GDDAT	
1477	023206	000403			BR	14\$	
1478	023210	163737	002732	002362	SUB	QMAX,GDDAT	
1479	023216	053777	002732	157030	BIS	QMAX,@RLDA	
1480	023224	012737	000001	002356	MOV	#MK,TMP1	
1481	023232	032777	000020	157014	BIT	#DAHS,@RLDA	
1482	023240	001003			BNE	5\$	
1483	023242	052737	000020	002356	BIS	#DAHS,TMP1	
1484	023250	004537	013420		JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
1485	023254	000006			SEEK		:SEEK
1486	023256	004537	014306		JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
1487	023262				CKLOOP		:CHECK IF /FL:LOE IS SET
	023262	104406			TRAP	C\$CLP1	
1488							
1489							
1490	023264	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1491	023270				CKLOOP		:CHECK IF /FL:LOE IS SET
	023270	104406			TRAP	C\$CLP1	
1492							
1493	023272	013777	002356	156754	MOV	TMP1,@RLDA	:SEND IN NEW DIFFERENCE WORD
1494	023300	004537	014306		JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
1495	023304				CKLOOP		:CHECK IF /FL:LOE IS SET
	023304	104406			TRAP	C\$CLP1	
1496							
1497	023306	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1498	023312				CKLOOP		:CHECK IF /FL:LOE IS SET
	023312	104406			TRAP	C\$CLP1	
1499							
1500	023314	004537	014220		JSR	R5,WTDYDY	:WAIT FOR DRIVE READY
1501	023320				CKLOOP		:CHECK IF /FL:LOE IS SET
	023320	104406			TRAP	C\$CLP1	
1502							
1503							
1504	023322	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1505	023326				CKLOOP		:CHECK IF /FL:LOE IS SET

```

023326 104406 TRAP CSCLP1
1506
1507 023330 004537 013420 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
1508 023334 000010 RDHDR ;READ HEADER
1509 023336 004537 014306 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
1510 023342 CKLOOP ;CHECK IF /FL:LOE IS SET
023342 104406 TRAP CSCLP1
1511
1512 023344 004537 013120 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
1513 023350 ESCAPE TST ;IF /FL:LOE SET LOOP, ELSE EXIT TST
023350 104410 TRAP C$ESCAPE
023352 000036 .WORD L10072-.
1514
1515 023354 013737 002314 002364 MOV E.MP,BDDAT ;READ HEADER
1516 023362 043737 002334 002364 BIC SECMSK,BDDAT ;CLEAR SECTOR ADDRESS
1517 023370 023737 002362 002364 CMP GDDAT,BDDAT ;IS HEADER CORRECT?
1518 023376 001404 BEQ 10$ ;IF SO BRANCH
1519
1520 023400 ERRDF 50.,EM57,ERR4
023400 104455 TRAP C$ERDF
023402 000062 .WORD 50
023404 006557 .WORD EM57
023406 010324 .WORD ERR4
1521 023410 10$:
1522
1523 023410 ENDTST ;****END OF TEST****
023410 L10072:
023410 104401 TRAP C$ETST
1524
1525
1526 .SBTTL **TEST 43** - EXTENSIVE CHECK OF HEADER CRC
1527
1528 023412 BGNTST ;****START OF TEST****
1529 023412 STARS
;*****
;MORE EXTENSIVE CHECK OF HEADER CRC. WE WILL SEEK
;AND READ HEADERS VERIFYING HDR CRC ACROSS THE
;PLATTER USING THE GROWING 0, GROWING 1, SHIFTING 0 AND
;GROWING 0 PATTERNS FOR TRACK ADDRESSES.
STARS
;*****
1530
1531
1532
1533
1534 023412
1535
1536
1537 023412 012703 002626 BGNSEG MOV #SKLST,R3 ;GET LIST OF DIFFERENCE WORDS
1538 023416 TRAP C$BSEG ;****START OF SEGMENT****
023416 104404
1539 023420 13:
1540 023420 004537 013420 JSR R5,LDFUNC ;ISSUE FUNCTION OF FOLLOWING WORD
1541 023424 000010 RDHDR ;READ HEADER
1542 023426 004537 014306 JSR R5,WTCRDY ;WAIT FOR CONTROLLER READY HIGH
1543 023432 CKLOOP ;CHECK IF /FL:LOE IS SET
023432 104406 TRAP CSCLP1
1544
1545 023434 004537 013120 JSR R5,CHERR ;CHECK CONTROLLER FOR ERRORS
1546 023440 CKLOOP ;CHECK IF /FL:LOE IS SET
023440 104406 TRAP CSCLP1
1547
  
```


1548	023442	013737	002314	002364	MOV	E.MP,BDDAT	:READ HEADER
1549	023450	043737	002334	002364	BIC	SECMASK,BDDAT	:CLEAR OUT SECTOR
1550	023456	001461			BEQ	5\$:IF ON TRACK ZERO, H.S. ZERO, OK
1551							
1552							:NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
1553							:ON ZERO.
1554							
1555	023460	042737	000100	002364	BIC	#RHHS,BDDAT	:CLEAR OUT HEAD SELECT
1556	023466	013777	002364	156560	MOV	BDDAT,@RLDA	:PUT CYLINDER AS DIFFERENCE WORD
1557	023474	052777	000001	156552	BIS	#MK,@RLDA	:SET MARKER BIT
1558	023502	004537	013420		JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
1559	023506	000006			SEEK		:SEEK
1560	023510	004537	014306		JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
1561	023514				CKLOOP		:CHECK IF /FL:LOE IS SET
	023514	104406			TRAP	C\$CLP1	
1562							
1563	023516	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1564	023522				CKLOOP		:CHECK IF /FL:LOE IS SET
	023522	104406			TRAP	C\$CLP1	
1565							
1566	023524	004537	014220		JSR	R5,WTDY	:WAIT FOR DRIVE READY
1567	023530				CKLOOP		:CHECK IF /FL:LOE IS SET
	023530	104406			TRAP	C\$CLP1	
1568							
1569	023532	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1570	023536				CKLOOP		:CHECK IF /FL:LOE IS SET
	023536	104406			TRAP	C\$CLP1	
1571							
1572							
1573	023540	004537	013420		JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
1574	023544	000010			RDHDR		:READ HEADER
1575	023546	004537	014306		JSR	R5,WTCRDY	:WAIT FOR CONTROLLER READY HIGH
1576	023552				CKLOOP		:CHECK IF /FL:LOE IS SET
	023552	104406			TRAP	C\$CLP1	
1577							
1578	023554	004537	013120		JSR	R5,CHERR	:CHECK CONTROLLER FOR ERRORS
1579	023560				CKLOOP		:CHECK IF /FL:LOE IS SET
	023560	104406			TRAP	C\$CLP1	
1580							
1581	023562	005037	002362		CLR	GDDAT	:CLEAR EXPECTED
1582	023566	013737	002364	002376	MOV	BDDAT,DWORD	:SAVE DIFFERENCE WORD
1583	023574	013737	002314	002364	MOV	E.MP,BDDAT	:READ HEADER
1584	023602	043737	002334	002364	BIC	SECMASK,BDDAT	:MASK OUT SECTOR BITS
1585	023610	001404			BEQ	5\$:BRANCH IF ON ZERO TRACK
1586							
1587	023612				ERRDF	51,EM54,ERR3	
	023612	104455			TRAP	C\$ERDF	
	023614	000063			.WORD	51	
	023616	006426			.WORD	EM54	
	023620	010252			.WORD	ERR3	
1588	023622				CKLOOP		:CHECK IF /FL:LOE IS SET
	023622	104406			TRAP	C\$CLP1	
1589							
1590	023624	011377	156424		MOV	(R3),@RLDA	:GET DIFFERENCE WORD
1591	023630	052777	000005	156416	BIS	#SIGN!MK,@RLDA	:SET SIGN (TOWARDS SPINDLE) AND MARKER
1592	023636	004537	013420		JSR	R5,LDFUNC	:ISSUE FUNCTION OF FOLLOWING WORD
1593	023642	000006			SEEK		:SEEK

1594	023644	004537	014306		JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY HIGH
1595	023650				CKLOOP			:CHECK IF /FL:LOE IS SET
	023650	104406			TRAP	C\$CLP1		
1596								
1597	023652	004537	013120		JSR	R5,CHERR		:CHECK CONTROLLER FOR ERRORS
1598	023656				CKLOOP			:CHECK IF /FL:LOE IS SET
	023656	104406			TRAP	C\$CLP1		
1599								
1600	023660	004537	014220		JSR	R5,WTD RDY		:WAIT FOR DRIVE READY
1601	023664				CKLOOP			:CHECK IF /FL:LOE IS SET
	023664	104406			TRAP	C\$CLP1		
1602								
1603								
1604	023666	004537	013120		JSR	R5,CHERR		:CHECK CONTROLLER FOR ERRORS
1605	023672				CKLOOP			:CHECK IF /FL:LOE IS SET
	023672	104406			TRAP	C\$CLP1		
1606								
1607	023674	004537	013420		JSR	R5,LDFUNC		:ISSUE FUNCTION OF FOLLOWING WORD
1608	023700	000010			RDHDR			:READ HEADER
1609	023702	004537	014306		JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY HIGH
1610	023706				CKLOOP			:CHECK IF /FL:LOE IS SET
	023706	104406			TRAP	C\$CLP1		
1611								
1612								
1613	023710	004537	013120		JSR	R5,CHERR		:CHECK CONTROLLER FOR ERRORS
1614	023714				CKLOOP			:CHECK IF /FL:LOE IS SET
	023714	104406			TRAP	C\$CLP1		
1615								
1616	023716	011337	002362		MOV	(R3),GDDAT		:GET EXPECTED CYLINDER
1617	023722	011337	002376	8\$:	MOV	(R3),DWORD		:SET UP DIFFERENCE FOR SEEK
1618	023726	013737	002314	002364	MOV	E.MP,BDDAT		:READ HEADER FROM RLMP
1619	023734	043737	002334	002364	BIC	SECMASK,BDDAT		:CLEAR OUT SECTOR BITS
1620	023742	023737	002362	002364	CMP	GDDAT,BDDAT		:DID SEEK GO TO THE RIGHT
1621	023750	001404			BEQ	9\$:TRACK, IF SO, GO GET NEXT
1622								
1623	023752				ERRDF	52,EM54,ERR3		
	023752	104455			TRAP	C\$ERDF		
	023754	000064			.WORD	52		
	023756	006426			.WORD	EM54		
	023760	010252			.WORD	ERR3		
1624	023762				9\$:	CKLOOP		:CHECK IF /FL:LOE IS SET
	023762	104406			TRAP	C\$CLP1		
1625								
1626	023764	013737	002314	024000	MOV	E.MP,10\$:GET HEADER WORD
1627	023772	004537	014032		JSR	R5,SIMBCC		:GO CALCULATE HEADER CRC
1628	023776	000020			16.			:16 BITS
1629	024000	000000		10\$:	.WORD	0		:HEADER GOES HERE
1630	024002	000000			.WORD	0		:START WITH ZERO CRC
1631	024004	013737	002344	024030	MOV	CALBCC,20\$		
1632	024012	013737	002316	024026	MOV	E.MP1,21\$		
1633	024020	004537	014032		JSR	R5,SIMBCC		
1634	024024	000020			16.			
1635	024026	000000		21\$:	.WORD	0		
1636	024030	000000		20\$:	.WORD	0		
1637	024032	013737	002344	002362	MOV	CALBCC,GDDAT		:MOVE CALCULATED CRC TO GDDAT
1638	024040	013737	002320	002364	MOV	E.MP2,BDDAT		:GET HEADER CRC FROM RLMP
1639	024046	023737	002362	002364	CMP	GDDAT,BDDAT		:IS CRC CORRECT?


```

1640 024054 001404          BEQ      11$          ;IF SO CONTINUE
1641
1642 024056          ERRDF    53.,EM42,ERR4
      024056 104455      TRAP    C$ERDF
      024060 000065      .WORD   53
      024062 006047      .WORD   EM42
      024064 010324      .WORD   ERR4
1643 024066          11$:    CKLOOP   ;CHECK IF /FL:LOE IS SET
      024066 104406      TRAP    C$CLP1
1644
1645
1646 024070 005723      TST     (R3)+          ;BUMP PATTERN
1647 024072 023727 002406 000001  CMP    T.DRIVE,#1
1648 024100 001005      BNE    2$
1649 024102 020327 002726  CMP    R3,#SKEND
1650 024106 001407      BEQ    12$
1651 024110 000137 023420  JMP    1$
1652 024114 020327 002770  2$:    CMP    R3,#SKEEND
1653 024120 001402      BEQ    12$
1654 024122 000137 023420  JMP    1$
1655 024126          12$:
1656
1657 024126          ENDSEG          ;****END OF SEGMENT****
      024126 10000$:
1658 024130          TRAP    C$ESEG          ;****END OF TEST****
      024130 104405
      024130          ENDTST
      024130 104401      L10073:  TRAP    C$ETST
1659
1660
1661          .SBTTL  **TEST 44** - VERIFY GET STATUS WHILE DRDY IS LOW
1662
1663 024132          BGNTST          ;****START OF TEST****
1664
1665 024132          STARS
      ;:*****
      ;:VERIFY THAT WE CAN ISSUE GET STATUS AND RECIEVE
      ;:THE STATUS WORD WHILE THE DRIVE IS IN NOTION SEEKING
      ;:STARS
      ;:*****
1666
1667
1668 024132          1$:
1669
1670
1671 024132          JSR     R5,LDFUNC          ;ISSUE FUNCTION OF FOLLOWING WORD
1672 024132 004537 013420  RDHDR          ;READ HEADER
1673 024136 000010          JSR     R5,WTCRDY          ;WAIT FOR CONTROLLER READY HIGH
1674 024140 004537 014306  CKLOOP        ;CHECK IF /FL:LOE IS SET
1675 024144          TRAP    C$CLP1
      024144 104406
1676
1677 024146 004537 013120  JSR     R5,CHERR          ;CHECK CONTROLLER FOR ERRORS
1678 024152          CKLOOP        ;CHECK IF /FL:LOE IS SET
      024152 104406      TRAP    C$CLP1
1679
1680 024154 013737 002314 002364  MOV    E.MP,BDDAT          ;READ HEADER
1681 024162 043737 002334 002364  BIC    SECM$K,BDDAT          ;CLEAR OUT SECTOR
1682 024170 001461          BEQ    5$
1683
      ;:IF ON TRACK ZERO, H.S. ZERO, OK
  
```

```

1684                                     ;NOT ON TRACK ZERO CALCULATE DIFFERENCE WORD AND PUT IT BACK
1685                                     ;ON ZERO.
1686
1687 024172 042737 000100 002364      BIC      #RHHS,BDDAT      ;CLEAR OUT HEAD SELECT
1688 024200 013777 002364 156046      MOV      BDDAT,@RLDA     ;PUT CYLINDER AS DIFFERENCE WORD
1689 024206 052777 000001 156040      BIS      #MK,@RLDA      ;SET MARKER BIT
1690 024214 004537 013420              JSR      R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1691 024220 000006              SEEK
1692 024222 004537 014306              JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY HIGH
1693 024226              CKLOOP
1693 024226 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1694
1695 024230 004537 013120              JSR      R5,CHERR       ;CHECK CONTROLLER FOR ERRORS
1696 024234              CKLOOP
1696 024234 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1697
1698 024236 004537 014220              JSR      R5,WTCRDY      ;WAIT FOR DRIVE READY
1699 024242              CKLOOP
1699 024242 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1700
1701 024244 004537 013120              JSR      R5,CHERR       ;CHECK CONTROLLER FOR ERRORS
1702 024250              CKLOOP
1702 024250 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1703
1704
1705 024252 004537 013420              JSR      R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1706 024256 000010              RDHDR
1707 024260 004537 014306              JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY HIGH
1708 024264              CKLOOP
1708 024264 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1709
1710 024266 004537 013120              JSR      R5,CHERR       ;CHECK CONTROLLER FOR ERRORS
1711 024272              CKLOOP
1711 024272 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1712
1713 024274 005037 002362              CLR      GDDAT          ;CLEAR EXPECTED
1714 024300 013737 002364 002376      MOV      BDDAT,DWORD    ;SAVE DIFFERENCE WORD
1715 024306 013737 002314 002364      MOV      E.MP,BDDAT     ;READ HEADER
1716 024314 043737 002334 002364      BIC      SECMSK,BDDAT   ;MASK OUT SECTOR BITS
1717 024322 001404              BEQ      5$             ;BRANCH IF ON ZERO TRACK
1718
1719 024324              ERRDF  54.,EM54,ERR3
1719 024324 104455              TRAP     C$ERDF
1719 024326 000066              .WORD   54
1719 024330 006426              .WORD   EM54
1719 024332 010252              .WORD   ERR3
1720 024334              5$: CKLOOP
1720 024334 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1721
1722 024336 012777 077601 155710      MOV      #77601,@RLDA   ;GET DIFFERENCE WORD
1723 024344 052777 000005 155702      BIS      #SIGN!MK,@RLDA ;SET SIGN (TOWARDS SPINDLE) AND MARKER
1724 024352 004537 013420              JSR      R5,LDFUNC       ;ISSUE FUNCTION OF FOLLOWING WORD
1725 024356 000006              SEEK
1726 024360 004537 014306              JSR      R5,WTCRDY      ;WAIT FOR CONTROLLER READY HIGH
1727 024364              CKLOOP
1727 024364 104406              TRAP     C$CLP1        ;CHECK IF /FL:LOE IS SET

1728
  
```


1729								
1730	024366	004537	013120		JSR	R5,CHERR		:CHECK CONTROLLER FOR ERRORS
1731	024372				CKLOOP			:CHECK IF /FL:LOE IS SET
	024372	104406			TRAP	C\$CLP1		
1732	024374	012777	000003	155652	MOV	#MK!GSBIT,@RLDA		
1733	024402	004537	013420		JSR	R5,LDFUNC		:ISSUE FUNCTION OF FOLLOWING WORD
1734	024406	000004			GSTAT			
1735	024410	004537	014306		JSR	R5,WTCRDY		:WAIT FOR CONTROLLER READY HIGH
1736	024414				CKLOOP			:CHECK IF /FL:LOE IS SET
	024414	104406			TRAP	C\$CLP1		
1737	024416	004537	013120		JSR	R5,CHERR		:CHECK CONTROLLER FOR ERRORS
1738								
1739	024422				ENDTST			:*****END OF TEST*****
	024422				L10074:			
	024422	104401			TRAP	C\$SETST		
1740								
1741	024424				BGNMOD	HRDPRM		
1742								
1743	024424				BGNHRD			
	024424	000032			.WORD	L10075-LSHARD/2		
1744								:WHAT TYPE OF CONTROLLER
1745								:RL11=1, RLV11=2, RLV12=3
1746	024426				GPRMD	CNTMSG,CNT,0,3,1,3,YES		
	024426	005032			.WORD	T\$CODE		
	024430	024526			.WORD	CNTMSG		
	024432	000003			.WORD	3		
	024434	000001			.WORD	T\$LOLIM		
	024436	000003			.WORD	T\$HILIM		
1747								:CONTROLLER BUS ADDRESS
1748	024440				GPRMA	CSRMSG,CSR,0,160000,177776,YES		
	024440	000031			.WORD	T\$CODE		
	024442	024512			.WORD	CSRMSG		
	024444	160000			.WORD	T\$LOLIM		
	024446	177776			.WORD	T\$HILIM		
1749								:INTERRUPT VECTOR
1750	024450				GPRMA	VECMMSG,VECT,0,0,776,YES		
	024450	001031			.WORD	T\$CODE		
	024452	024544			.WORD	VECMMSG		
	024454	000000			.WORD	T\$LOLIM		
	024456	000776			.WORD	T\$HILIM		
1751								:DRIVE NUMBER
1752	024460				GPRMD	DRMSG,DRBT,0,03400,0,7,YES		
	024460	004032			.WORD	T\$CODE		
	024462	024575			.WORD	DRMSG		
	024464	003400			.WORD	03400		
	024466	000000			.WORD	T\$LOLIM		
	024470	000007			.WORD	T\$HILIM		
1753								:DRIVE TYPE
1754	024472				GPRML	DRTYPE,TYPDR,1,YES		
	024472	003130			.WORD	T\$CODE		
	024474	024553			.WORD	DRTYPE		
	024476	000001			.WORD	1		
1755								:BREAK LEVEL
1756	024500				GPRMD	BRMSG,PRIOR,0,340,0,7,YES		
	024500	002032			.WORD	T\$CODE		
	024502	024533			.WORD	BRMSG		
	024504	000340			.WORD	340		

```

024506 000000 .WORD TSLOLIM
024510 000007 .WORD TSHILIM
1757
1758 024512 ENDHRD
.EVEN
024512 L10075:
1759
1760 024512 102 125 123 CSRMSG: .ASCIZ /BUS ADDRESS/
024515 040 101 104
024520 104 122 105
024523 123 123 000
1761 024526 122 114 061 CNTMSG: .ASCIZ /RL11/
024531 061 000
1762 024533 102 122 040 BRMSG: .ASCIZ /BR LEVEL/
024536 114 105 126
024541 105 114 000
1763 024544 126 105 103 VECMSG: .ASCIZ /VECTOR/
024547 124 117 122
024552 000
1764 024553 104 122 111 DRTYPE: .ASCIZ /DRIVE TYPE = RL01/
024556 126 105 040
024561 124 131 120
024564 105 040 075
024567 040 122 114
024572 060 061 000
1765 024575 104 122 111 DRMSG: .ASCIZ /DRIVE/
024600 126 105 000
.EVEN
1766
1767
1768 024604 ENDMOD
1769
1770
1771
1772 024604 BGNMOD SFTPRM
1773
1774 024604 BGNSFT
024604 000011 .WORD L10076-LSSOFT/2
1775 024606 GPRML DMSG,DLT,1,YES
024606 000130 .WORD TSCODE
024610 024630 .WORD DMSG
024612 000001 .WORD 1
1776 024614 XFERF 1$
024614 006044 .WORD TSCODE
1777 024616 GPRMD EMSG,ELT,0,177777,0,177777,YES
024616 001032 .WORD TSCODE
024620 024654 .WORD EMSG
024622 177777 .WORD 177777
024624 000000 .WORD TSLOLIM
024626 177777 .WORD TSHILIM
1778 024630 1$: ENDSFT
.EVEN
024630 L10076:
1779
1783
1784 024630 104 122 117 DMSG: .ASCIZ /DROP ON ERROR LIMIT/
1785 024654 105 122 122 EMSG: .ASCIZ /ERROR LIMIT/
1786
  
```



```
1790  
1791 .EVEN  
1792  
1793 024670 ENDMOD  
1794 024670 LASTAD  
024670 000000 .EVEN  
024672 000000 .WORD 0  
024674 L$LAST:: .WORD 0  
1795  
1796 000001 .END
```

ADDCOD 013030 G
 ADR = 000020 G
 AFREG 004644
 AFTER 013746
 ARLBA 004601
 ARLCS 004574
 ARLDA 004607
 ARLMP 004615
 ASSEMB= 000010
 BATEST 015114
 BA16 = 000020
 BA17 = 000040
 BCCFBK 002342
 BCSR 002262
 BDDAT 002364
 BEFORE 013676
 BEGPAT 002416
 BEREG 004623
 BGNTST 012430
 BIT0 = 000001 G
 BIT00 = 000001 G
 BIT01 = 000002 G
 BIT02 = 000004 G
 BIT03 = 000010 G
 BIT04 = 000020 G
 BIT05 = 000040 G
 BIT06 = 000100 G
 BIT07 = 000200 G
 BIT08 = 000400 G
 BIT09 = 001000 G
 BIT1 = 000002 G
 BIT10 = 002000 G
 BIT11 = 004000 G
 BIT12 = 010000 G
 BIT13 = 020000 G
 BIT14 = 040000 G
 BIT15 = 100000 G
 BIT2 = 000004 G
 BIT3 = 000010 G
 BIT4 = 000020 G
 BIT5 = 000040 G
 BIT6 = 000100 G
 BIT7 = 000200 G
 BIT8 = 000400 G
 BIT9 = 001000 G
 BOE = 000400 G
 BPRIOR 002264
 BRMSG 024533
 BVEC 002266
 B.BA 002274
 B.BE 002302
 B.CS 002272
 B.DA 002276
 B.MP 002300
 CALBCC 002344
 CHERR 013120
 CKERLT 013034

CLNCOD 012756 G
 CNT = 000012
 CNTMSG 024526
 COMP 004045
 CONT 012206
 CONTIN 012054
 CRDY = 000200
 CRTIM 004665
 CSEND 003070
 CSPAT 002772
 CSR = 000000
 CSRMSG 024512
 CTEST 014774
 CYLSK 002370
 CSAU = 000052
 CSAUTO= 000061
 CSBRK = 000022
 CSBSEG= 000004
 CSBSUB= 000002
 CSCCFG= 000045
 CSCCLK= 000062
 CSCLEA= 000012
 CSCLOS= 000035
 CSCLP1= 000006
 CSCVEC= 000036
 CSDCLN= 000044
 CSDODU= 000051
 CSDRPT= 000024
 CSDU = 000053
 CSEDIT= 000003
 CSERDF= 000055
 CSERHR= 000056
 CSERRO= 000060
 CSERSF= 000054
 CSERSO= 000057
 CSESCA= 000010
 CSESEG= 000005
 CSESUB= 000003
 CSETST= 000001
 CSEXIT= 000032
 CSGETB= 000026
 CSGETW= 000027
 CSGMAN= 000043
 CSGPHR= 000042
 CSGPLO= 000030
 CSGPRI= 000040
 CSINIT= 000011
 CSINLP= 000020
 CSMANI= 000050
 CSMEM = 000031
 CSMMSG = 000023
 CSOPEN= 000034
 CSPNTB= 000014
 CSPNTF= 000017
 CSPNTS= 000016
 CSPNTX= 000015
 CSQIO = 000377

CSRDBU= 000007
 CSREFG= 000047
 CSRESE= 000033
 CSREVI= 000003
 CSRFLA= 000021
 CSRPT = 000025
 CSSEFG= 000046
 CSSPRI= 000041
 CSSVEC= 000037
 CSTPRI= 000013
 DAHS = 000020
 DATEST 015220
 DCKMES 004026
 DEMES 003774
 DERFLG 002304
 DERR = 040000
 DIAGMC= 000000
 DLT = 000000
 DLTMES 004033
 DLYCNT 002414
 DMSG 024630
 DRBT = 000010
 DRDY = 000001
 DRIVE 002270
 DRMSG 024575
 DROP 011624
 DRPCOD 013024 G
 DRST = 000010
 DRTIM 004712
 DRTYPE 024553
 DRVRDY 012314
 DSPCOD 011632 G
 DS0 = 000000
 DS1 = 000400
 DS2 = 001000
 DS3 = 001400
 DWORD 002376
 EF.CON= 000036 G
 EF.NEW= 000035 G
 EF.PWR= 000034 G
 EF.RES= 000037 G
 EF.STA= 000040 G
 ELT = 000002
 EMSG 024654
 EM1 004740
 EM101 007723
 EM102 007770
 EM11 005211
 EM13 005252
 EM14 005304
 EM15 005332
 EM16 005360
 EM17 005406
 EM2 004765
 EM3 005012
 EM30 005441
 EM30A 005500

EM32 005540
 EM33 005574
 EM34 005641
 EM37 005716
 EM4 005037
 EM41 005756
 EM42 006047
 EM43 006105
 EM44 006204
 EM45 006237
 EM46 006272
 EM47 006325
 EM5 005064
 EM52 006356
 EM54 006426
 EM55 006465
 EM56 006520
 EM57 006557
 EM6 005135
 EM61 006660
 EM62 006741
 EM63 007024
 EM64 007105
 EM65 007170
 EM66 007251
 EM67 007334
 EM7 005163
 EM70 007371
 EM71 007426
 EM72 007463
 EM73 007516
 EM74 007551
 EM75 007603
 EM76 007635
 EM77 007670
 END 012470
 ENDPAT 002624
 ERCOUN 003074
 ERPOIN 003072
 ERR = 100000
 ERRVEC 002340
 ERRO 010160 G
 ERR1 010176 G
 ERR2 010210 G
 ERR3 010252 G
 ERR4 010324 G
 ERR5 010372 G
 ERR6 010404 G
 ERR7 010446 G
 EVL = 000004 G
 ESEND = 002100
 ESLOAD= 000035
 E.BA 002310
 E.BE 002322
 E.CS 002306
 E.DA 002312
 E.MP 002314

E.MP1 002316
 E.MP2 002320
 FIRST 002366
 FIX 013664
 FDNFNC 013636
 FRMT1 011024
 FRMT11 011303
 FRMT12 011344
 FRMT13 011423
 FRMT14 011454
 FRMT15 011520
 FRMT2 011064
 FRMT2A 011103
 FRMT2B 011116
 FRMT3 011132
 FRMT4 011137
 FRMT5 011175
 FRMT6 011246
 FRMT99 011172
 FSAU = 000015
 FSAUTO= 000020
 FSBGN = 000040
 FSCLEA= 000007
 FSDU = 000016
 FSEND = 000041
 FSHARD= 000004
 FSHW = 000013
 FSINIT= 000006
 FSJMP = 000050
 FSMOD = 000000
 FMSG = 000011
 FSPROT= 000021
 FSPWR = 000017
 FSRPT = 000012
 FSSEG = 000003
 FSSOFT= 000005
 FSSRV = 000010
 FSSUB = 000002
 FSSW = 000014
 FSTEST= 000001
 GDDAT 002362
 GLBDAT 002242 G
 GLBEQA 002242 G
 GLBERR 010160 G
 GLBSUB 013034 G
 GLBXT 003774 G
 GODRVR= 000202
 GSBIT = 000002
 GSTAT = 000004
 GSTINT 004535
 GSTMES 004476
 GSCNTO= 000200
 G\$DELM= 000372
 G\$DISP= 000003
 G\$EXCP= 000400
 G\$HILI= 000002
 G\$LOLI= 000001

SYMBOL TABLE

GSNO = 000000
 GSOFFS = 000400
 GSOF SI = 000376
 GSPRMA = 000001
 GSPRMD = 000002
 GSPRML = 000000
 GSRADA = 000140
 GSRADB = 000000
 GSRADD = 000040
 GSRADL = 000120
 GSRADO = 000020
 GSXFER = 000004
 GSYES = 000010
 HALMAX = 002704
 HCRCME = 004013
 HDRBUF = 003274
 HDRLST = 013640
 HMAX = 002734
 HNFMS = 004021
 HOE = 100000 G
 HPTCOD = 011604 G
 HRDPRM = 024424 G
 IBE = 010000 G
 IDU = 000040 G
 IER = 020000 G
 INITCO = 011772 G
 INTEN = 000100
 INTFLG = 002330
 INTSRV = 014212
 ISR = 000100 G
 IXE = 004000 G
 ISAU = 000041
 ISAUTO = 000041
 ISCLN = 000041
 ISDU = 000041
 ISHRD = 000041
 ISINIT = 000041
 ISMOD = 000041
 ISMSG = 000041
 ISPROT = 000040
 ISPTAB = 000041
 ISPWR = 000041
 ISRPT = 000041
 ISSEG = 000041
 ISSETU = 000041
 ISSFT = 000041
 ISSRV = 000041
 ISSUB = 000041
 ISTST = 000041
 JSJMP = 000167
 LDCSR = 002332
 LDFUNC = 013420
 LF = 004043
 LINE1 = 010504
 LINE2 = 010540
 LINE3 = 010752
 LOE = 040000 G

LOT = 000010 G
 LSACP = 002110 G
 LSAPT = 002036 G
 LSAU = 013030 G
 LSAUT = 002070 G
 LSAUTO = 012534 G
 LSCCP = 002106 G
 LSCLEA = 012756 G
 LSCO = 002032 G
 LSDEPO = 002011 G
 LSDESC = 002122 G
 LSDESP = 002076 G
 LSDEVP = 002060 G
 LSDISP = 011634 G
 LSDLY = 002116 G
 LSDTP = 002040 G
 LSDTYP = 002034 G
 LSDU = 013024 G
 LSDUT = 002072 G
 LSDVTY = 002230 G
 LSEF = 002052 G
 LSENV1 = 002044 G
 LSETP = 002102 G
 LSEXP1 = 002046 G
 LSEXP4 = 002064 G
 LSEXP5 = 002066 G
 LSHARD = 024426 G
 LSHIME = 002120 G
 LSHPCP = 002016 G
 LSHPTP = 002022 G
 LSHW = 011606 G
 LSICP = 002104 G
 LSINIT = 011772 G
 LSLADP = 002026 G
 LSLAST = 024674 G
 LSLoad = 002100 G
 LSLUN = 002074 G
 LSMREV = 002050 G
 LSNAME = 002000 G
 LSPRIO = 002042 G
 LSPROT = 011764 G
 LSPRT = 002112 G
 LSREPP = 002062 G
 LSREV = 002010 G
 LSSOFT = 024606 G
 LSSPC = 002056 G
 LSSPCP = 002020 G
 LSSPTP = 002024 G
 LSSTA = 002030 G
 LSSW = 011624 G
 LSTEST = 002114 G
 LSTIML = 002014 G
 LSUNIT = 002012 G
 L10000 = 010174
 L10001 = 010206
 L10002 = 010250
 L10003 = 010322

L10004 = 010370
 L10005 = 010402
 L10006 = 010444
 L10007 = 010502
 L10010 = 011622
 L10011 = 011632
 L10013 = 012532
 L10014 = 012754
 L10015 = 013022
 L10016 = 013026
 L10017 = 013032
 L10020 = 014216
 L10021 = 014500
 L10022 = 014574
 L10023 = 014670
 L10024 = 014764
 L10025 = 015104
 L10026 = 015210
 L10027 = 015276
 L10030 = 015422
 L10031 = 015546
 L10032 = 015654
 L10033 = 015754
 L10034 = 016044
 L10035 = 016144
 L10036 = 016254
 L10037 = 016330
 L10040 = 016366
 L10041 = 016512
 L10042 = 016652
 L10043 = 017012
 L10044 = 017216
 L10045 = 017250
 L10046 = 017456
 L10047 = 017544
 L10050 = 017712
 L10051 = 017742
 L10052 = 020114
 L10053 = 020202
 L10054 = 020330
 L10055 = 020352
 L10056 = 020432
 L10057 = 020576
 L10060 = 020734
 L10061 = 021252
 L10062 = 021346
 L10063 = 021412
 L10064 = 021536
 L10065 = 022154
 L10066 = 022306
 L10067 = 022450
 L10070 = 022610
 L10071 = 022762
 L10072 = 023410
 L10073 = 024130
 L10074 = 024422
 L10075 = 024512

L10076 = 024630
 MAXCYL = 002400
 MAXSEC = 002374
 MDHEDR = 002000 G
 MERLMT = 011626
 MK = 000001
 MSCRLF = 004040
 MXSEC1 = 002372
 NOOP0 = 000000
 NOOP7 = 000016
 NOPINT = 004157
 NOPMES = 004126
 NOPWR = 012014
 NXM = 020000
 NXMMES = 004001
 NXT = 012064
 OKHDR = 013650
 OPI = 002000
 OPIERR = 004053
 OPIMES = 004006
 OSAPTS = 000000
 OSAU = 000001
 OSBGNR = 000000
 OSBGNS = 000001
 OSDU = 000001
 OSERRT = 000000
 OSGNSW = 000001
 OSPOIN = 000001
 OSSETU = 000000
 PFLG = 002324
 PNT = 001000 G
 PRI = 002000 G
 PRIOR = 000004
 PRI00 = 000000 G
 PRI01 = 000040 G
 PRI02 = 000100 G
 PRI03 = 000140 G
 PRI04 = 000200 G
 PRI05 = 000240 G
 PRI06 = 000300 G
 PRI07 = 000340 G
 PWRFLG = 002242
 QMAX = 002732
 QUAMAX = 002702
 RDHDR = 000010
 READ = 000014
 REST = 012132
 RESTMS = 013402
 RHDINT = 004352
 RHMES = 004312
 RHHS = 000100
 RLBA = 002252
 RLBE = 002260
 RLCS = 002250
 RLDA = 002254
 RLMP = 002256
 RL2 = 002730

SECMSK = 002334
 SEEK = 000006
 SEKINT = 004444
 SEKMES = 004413
 SFTPRM = 024604 G
 SIGN = 000004
 SIMBCC = 014032
 SIZE = 000004
 SKEEND = 002770
 SKEND = 002726
 SKLST = 002626
 SPTCOD = 011622 G
 START = 012072
 START1 = 012034
 STHS = 000100
 SVCGBL = 000000
 SVCINS = 000000
 SVCSUB = 177777
 SVCTAG = 000000
 SVCTST = 177777
 SVHD = 002402
 SSLSYM = 010000
 TEMP2 = 002346
 TEMP3 = 002350
 TEMP4 = 002352
 TMPFNC = 002412
 TMP0 = 002354
 TMP1 = 002356
 TMP2 = 002360
 TRPFLG = 002326
 TRPHAN = 014204
 TYPDR = 000006
 TSARGC = 000001
 TSCODE = 001032
 TSERRN = 000066
 TSEXP = 000000
 TSFLAG = 000040
 TSGMAN = 000000
 TSHILI = 177777
 TSLAST = 000001
 TSLOLI = 000000
 TSLSYM = 010000
 TSLTNO = 000054
 TSNEST = 177777
 TSNS0 = 000000
 TSNS1 = 000005
 TSPTNU = 000000
 TSSAVL = 177777
 TSSEGL = 177777
 TSSEK0 = 010000
 TSSUBN = 000000
 TSTAGL = 177777
 TSTAGN = 010077
 TSTEMP = 000000
 TSTEST = 000054
 TSTSTM = 177777
 TSTSTS = 000001

SYMBOL TABLE

TSSAU = 010017	T1	014406 G	T24	017546 G	T39	022310 G	VECT = 000002
TSSAUT= 010014	T10	015550 G	T25	017714 G	T4	014672 G	WAITO 012332
TSSCLE= 010015	T11	015656 G	T26	017744 G	T40	022452 G	WAIT1 021302
TSSDU = 010016	T12	015756 G	T27	020116 G	T41	022612 G	WCKINT 004251
TSSHAR= 010075	T13	016046 G	T28	020204 G	T42	022764 G	WCKMES 004211
TSSHW = 010010	T14	016146 G	T29	020332 G	T43	023412 G	WHY 002404
TSSINI= 010013	T15	016256 G	T3	014576 G	T44	024132 G	WRCHK = 000002
TSSMSG= 010007	T16	016332 G	T30	020354 G	T5	014766 G	WRITE = 000012
TSSPRO= 010012	T17	016370 G	T31	020434 G	T6	015106 G	WTCRDY 014306
TSSSEG= 010000	T18	016514 G	T32	020600 G	T7	015212 G	WTDRDY 014220
TSSSOF= 010076	T19	016654 G	T33	020736 G	T8	015300 G	XPOLY 002336
TSSSRV= 010020	T2	014502 G	T34	021254 G	T9	015424 G	XXX 012114
TSSSW = 010011	T20	017014 G	T35	021350 G	UAM = 000200 G		XSALWA= 000000
TSSTES= 010074	T21	017220 G	T36	021414 G	UNITST 002246		XSFALS= 000040
T.CNTL 002410	T22	017252 G	T37	021540 G	UUT 002244		XSOFFS= 000400
T.DRIV 002406	T23	017460 G	T38	022156 G	VECMSG 024544		XSTRUE= 000020
T.SIZE 011630							

. ABS. 024674 000
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29184 WORDS (114 PAGES)
DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
A:CZRLGC.BIN,CZRLGC/C=SVC33.SRC,A:CZRLGC.MAC

BIT9	2-26#	2-38	2-39	2-211	2-271									
BOE	2-26#													
BPRIOR	2-87#	5-51*	8-648											
BRMSG	8-A56	8-A62#												
BVEC	2-88#	5-50*	5-97	5-154										
CSAU	2-7#	5-186												
CSAUTO	2-7#	5-141												
CSBRK	2-7#													
CSBSEG	2-7#	7-118	7-157	8-23	8-56	8-94	8-131	8-165	8-197	8-228	8-650	8-872	8-;34	8-?38
CSBSUB	2-7#													
CSCEFG	2-7#													
CSCLCK	2-7#													
CSCLEA	2-7#	5-164												
CSCLOS	2-7#													
CSCLP1	2-7#	7-22	7-48	7-73	7-98	8-360	8-403	8-447	8-491	8-500	8-535	8-563	8-574	8-583
	8-626	8-703	8-726	8-738	8-764	8-794	8-799	8-818	8-842	8-889	8-:05	8-:32	8-:55	8-:58
	8-:61	8-:92	8-:96	8-:02	8-:09	8-:39	8-:42	8-:57	8-:60	8-:63	8-:66	8-:71	8-:74	8-:83
	8-:90	8-:93	8-:96	8-:99	8-<05	8-<18	8-<55	8-<58	8-<61	8-<64	8-<69	8-01	8-04	8-07
	8-10	8-15	8-48	8-51	8-54	8-57	8-63	8-96	8-99	8->02	8->05	8->46	8->49	8->87
	8->91	8->95	8->98	8-?01	8-?05	8-?10	8-?43	8-?46	8-?61	8-?64	8-?67	8-?70	8-?76	8-?79
	8-?88	8-?95	8-?98	8-@01	8-@05	8-@10	8-@14	8-@24	8-@43	8-@75	8-@78	8-@93	8-@96	8-@99
	8-A02	8-A08	8-A11	8-A20	8-A27	8-A31	8-A36							
CSCVEC	2-7#	5-121	5-123	5-154	7-16	7-42	7-67	7-92						
CSDCLN	2-7#	5-90	6-18											
CSDODU	2-7#	5-89	5-128	5-139	6-17									
CSDRPT	2-7#													
CSDU	2-7#	5-176												
CSEDIT	2-7#	2-17												
CSERDF	2-7#	6-100	6-280	6-296	7-132	7-168	8-32	8-70	8-108	8-143	8-175	8-206	8-238	8-279
	8-304	8-327	8-359	8-368	8-402	8-412	8-446	8-455	8-490	8-499	8-507	8-572	8-581	8-590
	8-624	8-670	8-725	8-737	8-767	8-793	8-802	8-841	8-887	8-934	8-:04	8-:91	8-:08	8-:82
	8-<17	8-<79	8-26	8-73	8->21	8-?20	8-?87	8-@23	8-@42	8-A19				
CSEHR	2-7#													
CSEHRO	2-7#													
CSEHRSF	2-7#	7-21	7-47	7-72	7-97									
CSEHRSO	2-7#													
CSEHSCA	2-7#	7-133	7-169	8-33	8-71	8-109	8-144	8-176	8-207	8-239	8-566	8-659	8-662	8-672
	8-865	8-868	8-877	8-880	8-912	8-915	8-<08	8-<72	8-18	8-66	8->11	8->14	8-?13	
CSESEG	2-7#	7-140	8-7	8-40	8-78	8-114	8-149	8-181	8-212	8-244	8-678	8-893	8-<33	8-@57
CSESUB	2-7#													
CSETST	2-7#	7-23	7-49	7-74	7-99	7-141	8-8	8-41	8-79	8-115	8-150	8-182	8-213	8-245
	8-281	8-307	8-330	8-372	8-414	8-459	8-512	8-540	8-594	8-631	8-683	8-707	8-740	8-770
	8-805	8-821	8-846	8-894	8-937	8-:11	8-:35	8-:65	8-:11	8-<34	8-<82	8-29	8-76	8->24
	8-?23	8-@58	8-A39											
CSEXIT	2-7#													
CSGETB	2-7#													
CSGETW	2-7#													
CSGMAN	2-7#													
CSGPHR	2-7#	5-42												
CSGPLO	2-7#													
CSGPRI	2-7#													
CSINIT	2-7#	5-103												
CSINLP	2-7#	6-7												
CSMANI	2-7#													
CSMEM	2-7#													
CSMSG	2-7#	4-11	4-18	4-26	4-35	4-44	4-51	4-62	4-71					

LSAPT	2-17#				
LSAU	2-17	5-182#			
LSAUT	2-17#				
LSAUTO	2-17	5-108#			
LSCCP	2-17#				
LSCLEA	2-17	5-145#			
LSCO	2-17#				
LSDEPO	2-17#				
LSDESC	2-17	2-21#			
LSDESP	2-17#				
LSDEVP	2-17#				
LSDISP	2-17	4-144#			
LSDLY	2-17#	5-81	6-276	6-290	8-:29
LSDTP	2-17#				
LSDTYP	2-17#				
LSDU	2-17	5-172#			
LSDUT	2-17#				
LSDVTY	2-17	2-22#			
LSEF	2-17#				
LSEVI	2-17#				
LSETP	2-17#				
LSEXP1	2-17#				
LSEXP4	2-17#				
LSEXP5	2-17#				
LSHARD	2-17	8-A43	8-A43#		
LSHIME	2-17#				
LSHPCP	2-17#				
LSHPTP	2-17#				
LSHW	2-17	4-118	4-118#		
LSICP	2-17#				
LSINIT	2-17	5-12#			
LSLADP	2-17#				
LSLAST	2-17	8-A94#			
LSLOAD	2-17#				
LSLUN	2-17#				
LSMREV	2-17#				
LSNAME	2-17#				
LSPRIO	2-17#				
LSPROT	2-17	5-3#			
LSPRT	2-17#				
LSREPP	2-17#				
LSREV	2-17#				
LSSOFT	2-17	8-A74	8-A74#		
LSSPC	2-17#				
LSSPCP	2-17#				
LSSPTP	2-17#				
LSSTA	2-17#				
LSSW	2-17	4-132	4-132#		
LSTEST	2-17#				
LSTIML	2-17#				
LSUNIT	2-17#	5-17	5-36		
L1000	4-11#				
L1001	4-18#				
L1002	4-26#				
L1003	4-35#				
L1004	4-44#				

L10005	4-51#		
L10006	4-62#		
L10007	4-71#		
L10010	4-118	4-126#	
L10011	4-132	4-138#	
L10013	5-103#		
L10014	5-141#		
L10015	5-164#		
L10016	5-176#		
L10017	5-186#		
L10020	6-269#		
L10021	7-23#		
L10022	7-49#		
L10023	7-74#		
L10024	7-99#		
L10025	7-141#		
L10026	8-8#		
L10027	8-41#		
L10030	8-79#		
L10031	8-115#		
L10032	8-150#		
L10033	8-182#		
L10034	8-213#		
L10035	8-245#		
L10036	8-281#		
L10037	8-307#		
L10040	8-330#		
L10041	8-372#		
L10042	8-414#		
L10043	8-459#		
L10044	8-512#		
L10045	8-540#		
L10046	8-566	8-594#	
L10047	8-631#		
L10050	8-659	8-662	8-683#
L10051	8-707#		
L10052	8-740#		
L10053	8-770#		
L10054	8-805#		
L10055	8-821#		
L10056	8-845#		
L10057	8-865	8-868	8-894#
L10060	8-912	8-915	8-937#
L10061	8-:11#		
L10062	8-:35#		
L10063	8-:65#		
L10064	8-:11#		
L10065	8-<34#		
L10066	8-<72	8-<82#	
L10067	8-=18	8-=29#	
L10070	8-=66	8-=76#	
L10071	8->11	8->14	8->24#
L10072	8-?13	8-?23#	
L10073	8-@58#		
L10074	8-A39#		
L10075	8-A43	8-A58#	

7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-39	7-42	7-42
7-42	7-42	7-47	7-47	7-47	7-47	7-47	7-47	7-47	7-47	7-48	7-48	7-49	7-49
7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-64	7-67	7-67
7-67	7-67	7-72	7-72	7-72	7-72	7-72	7-72	7-72	7-72	7-73	7-73	7-74	7-74
7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-89	7-92	7-92
7-92	7-92	7-97	7-97	7-97	7-97	7-97	7-97	7-97	7-97	7-98	7-98	7-99	7-99
7-118	7-118	7-132	7-132	7-132	7-132	7-132	7-132	7-132	7-132	7-133	7-133	7-133	7-133
7-140	7-140	7-141	7-141	7-157	7-157	7-168	7-168	7-168	7-168	7-168	7-168	7-168	7-168
7-169	7-169	7-169	7-169	8-7	8-7	8-8	8-8	8-23	8-23	8-32	8-32	8-32	8-32
8-32	8-32	8-32	8-32	8-33	8-33	8-33	8-33	8-40	8-40	8-41	8-41	8-56	8-56
8-70	8-70	8-70	8-70	8-70	8-70	8-70	8-70	8-71	8-71	8-71	8-71	8-78	8-78
8-79	8-79	8-94	8-94	8-108	8-108	8-108	8-108	8-108	8-108	8-108	8-108	8-109	8-109
8-109	8-109	8-114	8-114	8-115	8-115	8-131	8-131	8-143	8-143	8-143	8-143	8-143	8-143
8-143	8-143	8-144	8-144	8-144	8-144	8-149	8-149	8-150	8-150	8-165	8-165	8-175	8-175
8-175	8-175	8-175	8-175	8-175	8-175	8-176	8-176	8-176	8-176	8-181	8-181	8-182	8-182
8-197	8-197	8-206	8-206	8-206	8-206	8-206	8-206	8-206	8-206	8-207	8-207	8-207	8-207
3-212	8-212	8-213	8-213	8-228	8-228	8-238	8-238	8-238	8-238	8-238	8-238	8-238	8-238
8-239	8-239	8-239	8-239	8-244	8-244	8-245	8-245	8-264	8-264	8-264	8-264	8-271	8-271
8-279	8-279	8-279	8-279	8-279	8-279	8-279	8-279	8-281	8-281	8-300	8-300	8-304	8-304
8-304	8-304	8-304	8-304	8-304	8-304	8-307	8-307	8-323	8-323	8-327	8-327	8-327	8-327
8-327	8-327	8-327	8-327	8-330	8-330	8-359	8-359	8-359	8-359	8-359	8-359	8-359	8-359
8-360	8-360	8-368	8-368	8-368	8-368	8-368	8-368	8-368	8-368	8-372	8-372	8-402	8-402
8-402	8-402	8-402	8-402	8-402	8-402	8-403	8-403	8-412	8-412	8-412	8-412	8-412	8-412
8-412	8-412	8-414	8-414	8-446	8-446	8-446	8-446	8-446	8-446	8-446	8-446	8-447	8-447
8-455	8-455	8-455	8-455	8-455	8-455	8-455	8-455	8-459	8-459	8-459	8-459	8-490	8-490
8-490	8-490	8-490	8-490	8-491	8-491	8-499	8-499	8-499	8-499	8-499	8-499	8-499	8-499
8-500	8-500	8-507	8-507	8-507	8-507	8-507	8-507	8-507	8-507	8-512	8-512	8-535	8-535
8-540	8-540	8-563	8-563	8-566	8-566	8-566	8-566	8-572	8-572	8-572	8-572	8-572	8-572
8-572	8-572	8-574	8-574	8-581	8-581	8-581	8-581	8-581	8-581	8-581	8-581	8-583	8-583
8-590	8-590	8-590	8-590	8-590	8-590	8-590	8-590	8-594	8-594	8-618	8-618	8-618	8-618
8-624	8-624	8-624	8-624	8-624	8-624	8-624	8-624	8-626	8-626	8-631	8-631	8-650	8-650
8-653	8-653	8-653	8-653	8-659	8-659	8-659	8-659	8-662	8-662	8-662	8-662	8-670	8-670
8-670	8-670	8-670	8-670	8-670	8-670	8-672	8-672	8-672	8-672	8-678	8-678	8-683	8-683
8-703	8-703	8-707	8-707	8-717	8-717	8-717	8-717	8-722	8-722	8-722	8-722	8-725	8-725
8-725	8-725	8-725	8-725	8-725	8-725	8-726	8-726	8-729	8-729	8-729	8-729	8-734	8-734
8-734	8-734	8-737	8-737	8-737	8-737	8-737	8-737	8-737	8-737	8-738	8-738	8-740	8-740
8-764	8-764	8-767	8-767	8-767	8-767	8-767	8-767	8-767	8-767	8-770	8-770	8-784	8-784
8-784	8-784	8-790	8-790	8-790	8-790	8-793	8-793	8-793	8-793	8-793	8-793	8-793	8-793
8-794	8-794	8-799	8-799	8-802	8-802	8-802	8-802	8-802	8-802	8-802	8-802	8-805	8-805
8-818	8-818	8-821	8-821	8-833	8-833	8-833	8-833	8-838	8-838	8-838	8-838	8-841	8-841
8-841	8-841	8-841	8-841	8-841	8-841	8-842	8-842	8-846	8-846	8-865	8-865	8-865	8-865
8-868	8-868	8-868	8-868	8-872	8-872	8-877	8-877	8-877	8-877	8-880	8-880	8-880	8-880
8-887	8-887	8-887	8-887	8-887	8-887	8-887	8-887	8-889	8-889	8-893	8-893	8-894	8-894
8-912	8-912	8-912	8-912	8-915	8-915	8-915	8-915	8-934	8-934	8-934	8-934	8-934	8-934
8-934	8-934	8-937	8-937	8-:04	8-:04	8-:04	8-:04	8-:04	8-:04	8-:04	8-:04	8-:05	8-:05
8-:11	8-:11	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29
8-:29	8-:29	8-:29	8-:29	8-:32	8-:32	8-:32	8-:32	8-:35	8-:35	8-:35	8-:35	8-:58	8-:58
8-:65	8-:65	8-:83	8-:83	8-:83	8-:83	8-:91	8-:91	8-:91	8-:91	8-:91	8-:91	8-:91	8-:91
8-:92	8-:92	8-:96	8-:96	8-:02	8-:02	8-:04	8-:04	8-:04	8-:04	8-:08	8-:08	8-:08	8-:08
8-:08	8-:08	8-:08	8-:08	8-:09	8-:09	8-:11	8-:11	8-:11	8-:11	8-:39	8-:39	8-:42	8-:42
8-:57	8-:57	8-:60	8-:60	8-:63	8-:63	8-:66	8-:66	8-:71	8-:71	8-:74	8-:74	8-:82	8-:82
8-:82	8-:82	8-:82	8-:82	8-:82	8-:82	8-:83	8-:83	8-:90	8-:90	8-:93	8-:93	8-:96	8-:96
8-:99	8-:99	8-<05	8-<05	8-<08	8-<08	8-<08	8-<08	8-<17	8-<17	8-<17	8-<17	8-<17	8-<17
8-<17	8-<17	8-<18	8-<18	8-<33	8-<33	8-<34	8-<34	8-<55	8-<55	8-<58	8-<58	8-<61	8-<61
8-<64	8-<64	8-<69	8-<69	8-<72	8-<72	8-<72	8-<72	8-<79	8-<79	8-<79	8-<79	8-<79	8-<79
8-<79	8-<79	8-<82	8-<82	8-01	8-01	8-01	8-01	8-04	8-04	8-07	8-07	8-10	8-10

TSSAUT	5-108#	5-141												
TSSCLE	5-145#	5-164												
TSSDU	5-172#	5-176												
TSSHAR	8-A43	8-A43#	8-A58											
TSSHW	4-118	4-118#	4-126											
TSSINI	5-12#	5-103												
TSSMSG	4-5#	4-11	4-13#	4-18	4-20#	4-26	4-28#	4-35	4-37#	4-44	4-46#	4-51	4-53#	4-62
	4-64#	4-71												
TSSPRO	5-3#													
TSSSEG	7-118	7-118#	7-133	7-140	7-140#	7-157	7-157#	7-169	8-7	8-7#	8-23	8-23#	8-33	8-40
	8-40#	8-56	8-56#	8-71	8-78	8-78#	8-94	8-94#	8-109	8-114	8-114#	8-131	8-131#	8-144
	8-149	8-149#	8-165	8-165#	8-176	8-181	8-181#	8-197	8-197#	8-207	8-212	8-212#	8-228	8-228#
	8-239	8-244	8-244#	8-650	8-650#	8-672	8-678	8-678#	8-872	8-872#	8-877	8-880	8-893	8-893#
	8-:34	8-:34#	8-<08	8-<33	8-<33#	8-?38	8-?38#	8-?57	8-?57#					
	8-A74	8-A74#	8-A78											
TSSSO	6-265#	6-269												
TSSSRV	4-132	4-132#	4-138											
TSSSW	7-3#	7-23	7-28#	7-49	7-54#	7-74	7-79#	7-99	7-104#	7-141	7-146#	8-8	8-13#	8-41
TSSSTES	8-46#	8-79	8-84#	8-115	8-120#	8-150	8-155#	8-182	8-187#	8-213	8-218#	8-245	8-250#	8-281
	8-286#	8-307	8-312#	8-330	8-335#	8-372	8-377#	8-414	8-419#	8-459	8-463#	8-512	8-516#	8-540
	8-545#	8-566	8-594	8-599#	8-631	8-636#	8-659	8-662	8-683	8-687#	8-707	8-712#	8-740	8-745#
	8-770	8-775#	8-805	8-809#	8-821	8-825#	8-846	8-851#	8-865	8-868	8-894	8-899#	8-912	8-915
	8-937	8-942#	8-:11	8-:16#	8-:35	8-:40#	8-:65	8-:70#	8-:11	8-:16#	8-<34	8-<39#	8-<72	8-<82
	8-<87#	8-=18	8-=29	8-=34#	8-=66	8-=76	8-=81#	8->11	8->14	8->24	8->29#	8-?13	8-?23	8-?28#
	8-?58	8-?63#	8-A39											
TSARGC	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17#	2-17#	2-17#
	2-17#	2-17#	2-17#	4-23	4-23	4-23	4-23	4-23#	4-23#	4-23#	4-23#	4-32	4-32	4-32
	4-32	4-32#	4-32#	4-32#	4-32#	4-41	4-41	4-41	4-41	4-41#	4-41#	4-41#	4-60	4-60
	4-60#	4-67	4-67	4-67	4-67#	4-67#	4-73	4-73	4-73	4-73	4-73#	4-73#	4-73#	4-76
	4-76	4-76	4-76	4-76	4-76	4-76	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#	4-77
	4-77	4-77	4-77	4-77	4-77#	4-77#	4-77#	4-77#	4-77#	4-77#	4-78	4-78	4-78	4-78
	4-78	4-78	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-79	4-79	4-79	4-79	4-79
	4-79#	4-79#	4-79#	4-79#	4-79#	4-82	4-82	4-82	4-82#	4-82#	4-83	4-83	4-83	4-83#
	4-83#	5-86	5-86	5-86#	5-126	5-126	5-126#	5-137	5-137	5-137#	6-15	6-15	6-15#	6-15#
TS	8-A46	8-A46	8-A46	8-A46#	8-A46#	8-A46#	8-A46	8-A48	8-A48	8-A48#	8-A48#	8-A48#	8-A48#	8-A50
CODE	8-A50	8-A50#	8-A50#	8-A50#	8-A52	8-A52	8-A52	8-A52#	8-A52#	8-A52#	8-A52#	8-A54	8-A54	8-A54#
	8-A54#	8-A54#	8-A56	8-A56	8-A56	8-A56#	8-A56#	8-A56#	8-A56#	8-A75	8-A75	8-A75	8-A75#	8-A75#
	8-A76	8-A76	8-A76	8-A76	8-A76	8-A76	8-A76#	8-A76#	8-A76#	8-A76#	8-A76#	8-A77	8-A77	8-A77#
	8-A77#	8-A77#												
TSERRN	2-7#	6-100	6-100#	6-280	6-280#	6-296	6-296#	7-21	7-21#	7-47	7-47#	7-72	7-72#	7-97
	7-97#	7-132	7-132#	7-168	7-168#	8-32	8-32#	8-70	8-70#	8-108	8-108#	8-143	8-143#	8-175
	8-175#	8-206	8-206#	8-238	8-238#	8-279	8-279#	8-304	8-304#	8-327	8-327#	8-359	8-359#	8-368
	8-368#	8-402	8-402#	8-412	8-412#	8-446	8-446#	8-455	8-455#	8-490	8-490#	8-499	8-499#	8-507
	8-507#	8-572	8-572#	8-581	8-581#	8-590	8-590#	8-624	8-624#	8-670	8-670#	8-725	8-725#	8-737
	8-737#	8-767	8-767#	8-793	8-793#	8-802	8-802#	8-841	8-841#	8-887	8-887#	8-934	8-934#	8-:04
	8-:04#	8-:91	8-:91#	8-:08	8-:08#	8-:82	8-:82#	8-<17	8-<17#	8-<79	8-<79#	8-=26	8-=26#	8-=73
	8-=73#	8->21	8->21#	8-?20	8-?20#	8-?87	8-?87#	8-?23	8-?23#	8-?42	8-?42#	8-A19	8-A19#	
	8-A46	8-A46#	8-A48	8-A48#	8-A50	8-A50#	8-A52	8-A52#	8-A56	8-A56#	8-A77	8-A77#	8-A77#	
TSEXCP	7-133	7-133#	7-133#	7-169	7-169#	7-169#	8-33	8-33#	8-33#	8-71	8-71#	8-71#	8-71#	8-109
TSFLAG	8-109#	8-144	8-144#	8-144#	8-176	8-176#	8-176#	8-176#	8-207	8-207#	8-207#	8-239	8-239#	8-109#
	8-566#	8-566#	8-659	8-659#	8-659#	8-662	8-662#	8-662#	8-672	8-672#	8-672#	8-672#	8-672#	8-566
	8-868	8-868#	8-868#	8-877	8-877#	8-877#	8-880	8-880#	8-880#	8-912	8-912#	8-912#	8-912#	8-865#
	8-915#	8-<08	8-<08#	8-<08#	8-<72	8-<72#	8-<72#	8-<72#	8-=18	8-=18#	8-=18#	8-=66	8-=66#	8-865#
	8->11#	8->11#	8->14	8->14#	8->14#	8-?13	8-?13#	8-?13#						8-915#
TSGMAN	2-7#													8->11
TSHILI	8-A46	8-A46#	8-A48	8-A48#	8-A50	8-A50#	8-A52	8-A52#	8-A56	8-A56#	8-A77	8-A77#	8-A77#	

TSLAST	2-7#	8-A94#													
TSLOLI	8-A46	8-A46#	8-A48	8-A48#	8-A50	8-A50#	8-A52	8-A52#	8-A56	8-A56#	8-A77	8-A77#	5-103	5-141	
TSLSYM	2-7	2-7#	4-11	4-18	4-26	4-35	4-44	4-51	4-62	4-71	4-126	4-138	8-115	8-150	
	5-164	5-176	5-186	6-269	7-23	7-49	7-74	7-99	7-141	8-8	8-41	8-79	8-631	8-683	
	8-182	8-213	8-245	8-281	8-307	8-330	8-372	8-414	8-459	8-512	8-540	8-594	8-631	8-683	
	8-707	8-740	8-770	8-805	8-821	8-846	8-894	8-937	8-:11	8-:35	8-:65	8-:11	8-<34	8-<82	
	8-=29	8-=76	8->24	8-?23	8-?58	8-A39	8-A58	8-A78							
TSLTNO	8-A94#														
TSNEST	2-7#	2-15	2-15	2-15#	2-19	2-19	2-19	2-19#	2-24	2-24	2-24#	2-72	2-72	2-72	
	2-72#	2-74	2-74	2-74#	2-298	2-298	2-298	2-298#	3-1	3-1	3-1#	3-91	3-91	3-91	
	3-91#	4-3	4-3	4-3#	4-5	4-5	4-5#	4-11	4-11	4-11	4-11#	4-13	4-13	4-13#	
	4-18	4-18	4-18	4-18#	4-20	4-20	4-20#	4-26	4-26	4-26	4-26#	4-28	4-28	4-28#	
	4-35	4-35	4-35	4-35#	4-37	4-37	4-37#	4-44	4-44	4-44	4-44#	4-46	4-46	4-46#	
	4-51	4-51	4-51	4-51#	4-53	4-53	4-53#	4-62	4-62	4-62	4-62#	4-64	4-64	4-64#	
	4-71	4-71	4-71	4-71#	4-114	4-114	4-114	4-114#	4-116	4-116	4-116#	4-118	4-118	4-118#	
	4-126	4-126	4-126	4-126#	4-128	4-128	4-128	4-128#	4-130	4-130	4-130#	4-132	4-132	4-132#	
	4-138	4-138	4-138	4-138#	4-140	4-140	4-140	4-140#	4-142	4-142	4-142#	4-146	4-146	4-146#	
	4-146#	5-3	5-3	5-3#	5-7	5-7	5-7	5-7#	5-10	5-10	5-10#	5-12	5-12	5-12#	
	5-103	5-103	5-103	5-103#	5-105	5-105	5-105	5-105#	5-108	5-108	5-108#	5-141	5-141	5-141#	
	5-141#	5-143	5-143	5-143#	5-145	5-145	5-145#	5-164	5-164	5-164	5-164#	5-166	5-166	5-166#	
	5-166#	5-170	5-170	5-170#	5-172	5-172	5-172#	5-176	5-176	5-176	5-176#	5-178	5-178	5-178#	
	5-178#	5-180	5-180	5-180#	5-182	5-182	5-182#	5-186	5-186	5-186	5-186#	5-188	5-188	5-188#	
	5-188#	6-4	6-4	6-4#	6-265	6-265	6-265#	6-269	6-269	6-269	6-269#	6-304	6-304	6-304#	
	6-304#	7-3	7-3	7-3#	7-23	7-23	7-23#	7-28	7-28	7-28	7-28#	7-49	7-49	7-49#	
	7-49#	7-54	7-54	7-54#	7-74	7-74	7-74#	7-79	7-79	7-79	7-79#	7-99	7-99	7-99#	
	7-99#	7-104	7-104	7-104#	7-118	7-118	7-118#	7-140	7-140	7-140	7-140#	7-141	7-141	7-141#	
	7-141#	7-146	7-146	7-146#	7-157	7-157	7-157#	8-7	8-7	8-7	8-7#	8-8	8-8	8-8#	
	8-8#	8-13	8-13	8-13#	8-23	8-23	8-23#	8-40	8-40	8-40	8-40#	8-41	8-41	8-41#	
	8-41#	8-46	8-46	8-46#	8-56	8-56	8-56#	8-78	8-78	8-78	8-78#	8-79	8-79	8-79#	
	8-79#	8-84	8-84	8-84#	8-94	8-94	8-94#	8-114	8-114	8-114	8-114#	8-115	8-115	8-115#	
	8-115#	8-120	8-120	8-120#	8-131	8-131	8-131#	8-149	8-149	8-149	8-149#	8-150	8-150	8-150#	
	8-150#	8-155	8-155	8-155#	8-165	8-165	8-165#	8-181	8-181	8-181	8-181#	8-182	8-182	8-182#	
	8-182#	8-187	8-187	8-187#	8-197	8-197	8-197#	8-212	8-212	8-212	8-212#	8-213	8-213	8-213#	
	8-213#	8-218	8-218	8-218#	8-228	8-228	8-228#	8-244	8-244	8-244	8-244#	8-245	8-245	8-245#	
	8-245#	8-250	8-250	8-250#	8-281	8-281	8-281#	8-286	8-286	8-286	8-286#	8-307	8-307	8-307#	
	8-307#	8-312	8-312	8-312#	8-330	8-330	8-330#	8-335	8-335	8-335	8-335#	8-372	8-372	8-372#	
	8-372#	8-377	8-377	8-377#	8-414	8-414	8-414#	8-419	8-419	8-419	8-419#	8-459	8-459	8-459#	
	8-459#	8-463	8-463	8-463#	8-512	8-512	8-512#	8-516	8-516	8-516	8-516#	8-540	8-540	8-540#	
	8-540#	8-545	8-545	8-545#	8-594	8-594	8-594#	8-599	8-599	8-599	8-599#	8-631	8-631	8-631#	
	8-631#	8-636	8-636	8-636#	8-650	8-650	8-650#	8-678	8-678	8-678	8-678#	8-683	8-683	8-683#	
	8-683#	8-687	8-687	8-687#	8-707	8-707	8-707#	8-712	8-712	8-712	8-712#	8-740	8-740	8-740#	
	8-740#	8-745	8-745	8-745#	8-770	8-770	8-770#	8-775	8-775	8-775	8-775#	8-805	8-805	8-805#	
	8-805#	8-809	8-809	8-809#	8-821	8-821	8-821#	8-825	8-825	8-825	8-825#	8-846	8-846	8-846#	
	8-846#	8-851	8-851	8-851#	8-872	8-872	8-872#	8-893	8-893	8-893	8-893#	8-894	8-894	8-894#	
	8-894#	8-899	8-899	8-899#	8-937	8-937	8-937#	8-942	8-942	8-942#	8-:11	8-:11	8-:11	8-:11#	
	8-:11#	8-:16	8-:16	8-:16#	8-:35	8-:35	8-:35#	8-:40	8-:40	8-:40#	8-:65	8-:65	8-:65	8-:65#	
	8-:65#	8-:70	8-:70	8-:70#	8-:11	8-:11	8-:11#	8-:16	8-:16	8-:16#	8-:34	8-:34	8-:34#	8-:34#	
	8-<33	8-<33	8-<33	8-<33#	8-<34	8-<34	8-<34#	8-<39	8-<39	8-<39#	8-<82	8-<82	8-<82	8-<82#	
	8-<82#	8-<87	8-<87	8-<87#	8-=29	8-=29	8-=29#	8-=34	8-=34	8-=34#	8-=76	8-=76	8-=76	8-=76#	
	8-=76#	8-=81	8-=81	8-=81#	8->24	8->24	8->24#	8->29	8->29	8->29#	8-?23	8-?23	8-?23	8-?23#	
	8-?23#	8-?28	8-?28	8-?28#	8-?38	8-?38	8-?38#	8-?57	8-?57	8-?57#	8-?58	8-?58	8-?58	8-?58#	
	8-?58#	8-?63	8-?63	8-?63#	8-A39	8-A39	8-A39#	8-A41	8-A41	8-A41#	8-A43	8-A43	8-A43#	8-A43#	
	8-A58	8-A58	8-A58	8-A58#	8-A68	8-A68	8-A68#	8-A72	8-A72	8-A72#	8-A74	8-A74	8-A74	8-A74#	
	8-A76	8-A78	8-A78	8-A78#	8-A93	8-A93	8-A93#	8-A93#	8-A93#	8-A93#	8-A93#	8-A93#	8-A93#	8-A93#	
TSNSO	2-15#	2-19	2-24#	2-72	2-74#	2-298	3-1#	3-91	4-3#	4-114	4-116#	4-128	4-130#	4-140	
	4-142#	4-146	5-3#	5-7	5-10#	5-105	5-108#	5-141	5-143#	5-166	5-170#	5-178	5-180#	5-188	

4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-146
4-146#	5-7	5-7#	5-103	5-103#	5-105	5-105#	5-141	5-141#	5-164	5-164#	5-166	5-166#	5-176
5-176#	5-178	5-178#	5-186	5-186#	5-188	5-188#	6-269	6-269#	6-304	6-304#	7-4	7-4#	7-9
7-9#	7-23	7-23#	7-31	7-31#	7-36	7-36#	7-49	7-49#	7-55	7-55#	7-60	7-60#	7-74
7-74#	7-80	7-80#	7-85	7-85#	7-99	7-99#	7-108	7-108#	7-113	7-113#	7-133	7-133#	7-133#
7-140	7-140#	7-141	7-141#	7-148	7-148#	7-153	7-153#	7-169	7-169#	7-169#	8-7	8-7#	8-8
8-8#	8-15	8-15#	8-19	8-19#	8-33	8-33#	8-33#	8-40	8-40#	8-41	8-41#	8-47	8-47#
8-52	8-52#	8-71	8-71#	8-71#	8-78	8-78#	8-79	8-79#	8-86	8-86#	8-90	8-90#	8-109
8-109#	8-109#	8-114	8-114#	8-115	8-115#	8-122	8-122#	8-127	8-127#	8-144	8-144#	8-144#	8-149
8-149#	8-150	8-150#	8-157	8-157#	8-161	8-161#	8-176	8-176#	8-176#	8-181	8-181#	8-182	8-182#
8-189	8-189#	8-193	8-193#	8-207	8-207#	8-207#	8-212	8-212#	8-213	8-213#	8-220	8-220#	8-224
8-224#	8-239	8-239#	8-239#	8-244	8-244#	8-245	8-245#	8-252	8-252#	8-261	8-261#	8-281	8-281#
8-288	8-288#	8-292	8-292#	8-307	8-307#	8-314	8-314#	8-318	8-318#	8-330	8-330#	8-337	8-337#
8-343	8-343#	8-372	8-372#	8-378	8-378#	8-384	8-384#	8-414	8-414#	8-422	8-422#	8-428	8-428#
8-459	8-459#	8-466	8-466#	8-471	8-471#	8-512	8-512#	8-520	8-520#	8-525	8-525#	8-540	8-540#
8-547	8-547#	8-550	8-550#	8-566	8-566#	8-594	8-594#	8-601	8-601#	8-611	8-611#	8-631	8-631#
8-638	8-638#	8-642	8-642#	8-659	8-659#	8-662	8-662#	8-672	8-672#	8-672#	8-678	8-678#	8-683
8-683#	8-690	8-690#	8-696	8-696#	8-707	8-707#	8-740	8-740#	8-747	8-747#	8-753	8-753#	8-770
8-770#	8-776	8-776#	8-781	8-781#	8-805	8-805#	8-810	8-810#	8-813	8-813#	8-821	8-821#	8-827
8-827#	8-830	8-830#	8-846	8-846#	8-854	8-854#	8-858	8-858#	8-865	8-865#	8-868	8-868#	8-877
8-877#	8-877#	8-880	8-880#	8-880#	8-893	8-893#	8-894	8-894#	8-901	8-901#	8-905	8-905#	8-912
8-912#	8-915	8-915#	8-937	8-937#	8-945	8-945#	8-951	8-951#	8-:11	8-:11#	8-:17	8-:17#	8-:21
8-:21#	8-:35	8-:35#	8-:43	8-:43#	8-:47	8-:47#	8-:65	8-:65#	8-:73	8-:73#	8-:77	8-:77#	8-:11
8-:11#	8-:21	8-:21#	8-:30	8-:30#	8-<08	8-<08#	8-<08#	8-<33	8-<33#	8-<34	8-<34#	8-<43	8-<43#
8-<46	8-<46#	8-<72	8-<72#	8-<82	8-<82#	8-<90	8-<90#	8-<93	8-<93#	8-=18	8-=18#	8-=29	8-=29#
8-=36	8-=36#	8-=40	8-=40#	8-=66	8-=66#	8-=76	8-=76#	8-=83	8-=83#	8-=87	8-=87#	8->11	8->11#
8->14	8->14#	8->24	8->24#	8->32	8->32#	8->40	8->40#	8-?13	8-?13#	8-?23	8-?23#	8-?29	8-?29#
8-?34	8-?34#	8-a57	8-a57#	8-a58	8-a58#	8-a65	8-a65#	8-a68	8-a68#	8-A39	8-A39#	8-A46	8-A46#
8-A46	8-A46#	8-A46#	8-A46#	8-A48	8-A48#	8-A48	8-A48#	8-A48#	8-A48#	8-A50	8-A50#	8-A50	8-A50#
8-A50#	8-A50#	8-A52	8-A52#	8-A52	8-A52#	8-A52#	8-A52#	8-A54	8-A54#	8-A54	8-A54#	8-A54#	8-A54#
8-A56	8-A56	8-A56	8-A56#	8-A56#	8-A56#	8-A58	8-A58#	8-A68	8-A68#	8-A75	8-A75#	8-A75	8-A75#
8-A75#	8-A75#	8-A77	8-A77#	8-A77	8-A77#	8-A77#	8-A77#	8-A78	8-A78#	8-A93	8-A93#	8-A93	8-A93#
2-7#	7-3	7-3	7-3#	7-28	7-28#	7-28#	7-54	7-54	7-54#	7-79	7-79	7-79#	7-104
7-104	7-104#	7-146	7-146#	7-146#	8-13	8-13	8-13#	8-46	8-46	8-46#	8-84	8-84#	8-84#
8-120	8-120	8-120#	8-155	8-155#	8-187	8-187	8-187#	8-187#	8-218	8-218	8-218#	8-250	8-250
8-250#	8-286	8-286	8-286#	8-312	8-312	8-312#	8-335	8-335	8-335#	8-377	8-377	8-377#	8-419
8-419	8-419#	8-463	8-463#	8-463#	8-516	8-516#	8-545	8-545	8-545#	8-599	8-599	8-599#	8-599#
8-636	8-636	8-636#	8-687	8-687	8-687#	8-712	8-712	8-712#	8-745	8-745	8-745#	8-775	8-775
8-775#	8-809	8-809	8-809#	8-825	8-825	8-825#	8-851	8-851	8-851#	8-899	8-899	8-899#	8-942
8-942	8-942#	8-:16	8-:16#	8-:16#	8-:40	8-:40#	8-:70	8-:70	8-:70#	8-:70#	8-:16	8-:16#	8-:16#
8-<39	8-<39	8-<39#	8-<87	8-<87#	8-<87#	8-=34	8-=34	8-=34#	8-=81	8-=81	8-=81#	8->29	8->29
8->29#	8-?28	8-?28	8-?28#	8-a63	8-a63	8-a63#	8-A94						
2-7#	4-11	4-18	4-23	4-26	4-32	4-35	4-41	4-44	4-51	4-60	4-62	4-67	4-71
4-73	4-76	4-77	4-78	4-79	4-82	4-83	5-14	5-15	5-19	5-21	5-30	5-42	5-86
5-89	5-90	5-97	5-99	5-103	5-112	5-121	5-126	5-128	5-137	5-139	5-141	5-147	5-154
5-164	5-176	5-186	6-7	6-15	6-17	6-18	6-100	6-280	6-296	7-13	7-16	7-21	7-22
7-23	7-39	7-42	7-47	7-48	7-49	7-64	7-67	7-72	7-73	7-74	7-89	7-92	7-97
7-98	7-99	7-118	7-132	7-133	7-140	7-141	7-157	7-168	7-169	8-7	8-8	8-23	8-32
8-33	8-40	8-41	8-56	8-70	8-71	8-78	8-79	8-94	8-108	8-109	8-114	8-115	8-131
8-143	8-144	8-149	8-150	8-165	8-175	8-176	8-181	8-182	8-197	8-206	8-207	8-212	8-213
8-228	8-238	8-239	8-244	8-245	8-264	8-271	8-279	8-281	8-300	8-304	8-307	8-323	8-327
8-330	8-359	8-360	8-368	8-372	8-402	8-403	8-412	8-414	8-446	8-447	8-455	8-459	8-490
8-491	8-499	8-500	8-507	8-512	8-535	8-540	8-563	8-566	8-572	8-574	8-581	8-583	8-590
8-594	8-618	8-624	8-626	8-631	8-650	8-653	8-659	8-662	8-670	8-672	8-678	8-683	8-703
8-707	8-717	8-722	8-725	8-726	8-729	8-734	8-737	8-738	8-740	8-764	8-767	8-770	8-784
8-790	8-793	8-794	8-799	8-802	8-805	8-818	8-821	8-833	8-838	8-841	8-842	8-846	8-865

TSTEST

TSTSTM

	8-868	3-872	8-877	8-880	8-887	8-889	8-893	8-894	8-912	8-915	8-934	8-937	8-:04	8-:05
	8-:11	8-:32	8-:35	8-:55	8-:58	8-:61	8-:65	8-:83	8-:91	8-:92	8-:96	8-:02	8-:04	8-:08
	8-:09	8-:11	8-:34	8-:39	8-:42	8-:57	8-:60	8-:63	8-:66	8-:71	8-:74	8-:82	8-:83	8-:90
	8-:93	8-:96	8-:99	8-<05	8-<08	8-<17	8-<18	8-<33	8-<34	8-<55	8-<58	8-<61	8-<64	8-<69
	8-<72	8-<79	8-<82	8-=01	8-=04	8-=07	8-=10	8-=15	8-=18	8-=26	8-=29	8-=48	8-=51	8-=54
	8-=57	8-=63	8-=66	8-=73	8-=76	8-=96	8-=99	8->02	8->05	8->11	8->14	8->21	8->24	8->46
	8->49	8->87	8->91	8->95	8->98	8-?01	8-?05	8-?10	8-?13	8-?20	8-?23	8-?38	8-?43	8-?46
	8-?61	8-?64	8-?67	8-?70	8-?76	8-?79	8-?87	8-?88	8-?95	8-?98	8-@01	8-@05	8-@10	8-@14
	8-@23	8-@24	8-@42	8-@43	8-@57	8-@58	8-@75	8-@78	8-@93	8-@96	8-@99	8-A02	8-A08	8-A11
	8-A19	8-A20	8-A27	8-A31	8-A36	8-A39								
T\$TSTS	2-7#	7-3#	7-28#	7-54#	7-79#	7-104#	7-146#	8-13#	8-46#	8-84#	8-120#	8-155#	8-187#	8-218#
	8-250#	8-286#	8-312#	8-335#	8-377#	8-419#	8-463#	8-516#	8-545#	8-599#	8-636#	8-687#	8-712#	8-745#
	8-775#	8-809#	8-825#	8-851#	8-899#	8-942#	8-:16#	8-:40#	8-:70#	8-:16#	8-<39#	8-<87#	8-=34#	8-=81#
	8->29#	8-?28#	8-@63#											
T.CNTL	2-130#	5-54*	5-64	6-182	6-198	7-160	8-135	8-296	8-528	8-552	8-614	8-644		
T.DRIV	2-129#	5-52*	8-<21	8->59	8-@47									
T.SIZE	4-136#													
T1	4-144	7-3#												
T10	4-144	8-120#												
T11	4-144	8-155#												
T12	4-144	8-187#												
T13	4-144	8-218#												
T14	4-144	8-250#												
T15	4-144	8-286#												
T16	4-144	8-312#												
T17	4-144	8-335#												
T18	4-144	8-377#												
T19	4-144	8-419#												
T2	4-144	7-28#												
T20	4-144	8-463#												
T21	4-144	8-516#												
T22	4-144	8-545#												
T23	4-144	8-599#												
T24	4-144	8-636#												
T25	4-144	8-687#												
T26	4-144	8-712#												
T27	4-144	8-745#												
T28	4-144	8-775#												
T29	4-144	8-809#												
T3	4-144	7-54#												
T30	4-144	8-825#												
T31	4-144	8-851#												
T32	4-144	8-899#												
T33	4-144	8-942#												
T34	4-144	8-:16#												
T35	4-144	8-:40#												
T36	4-144	8-:70#												
T37	4-144	8-:16#												
T38	4-144	8-<39#												
T39	4-144	8-<87#												
T4	4-144	7-79#												
T40	4-144	8-=34#												
T41	4-144	8-=81#												
T42	4-144	8->29#												
T43	4-144	8-?28#												
T44	4-144	8-@63#												

CZ
CR

MS

MS
MA
ME
OP
PO
PR
PR
PR
RE
RE
SE
SE
SL
ST

SV
XF
XF
XF

ENDHW	1-463#	2-7#	4-126											
ENDINI	1-473#	2-7#	5-103											
ENDMOD	1-485#	2-7#	2-19	2-72	2-298	3-91	4-114	4-128	4-140	4-146	5-105	5-166	5-178	5-188
	6-304	8-A68	8-A93											
ENDMSG	1-498#	2-7#	4-11	4-18	4-26	4-35	4-44	4-51	4-62	4-71				
ENDPRO	1-510#	2-7#	5-7											
ENDPTA	1-518#	2-7#												
ENDRPT	1-527#	2-7#												
ENDSEG	1-539#	2-7#	7-140	8-7	8-40	8-78	8-114	8-149	8-181	8-212	8-244	8-678	8-893	8-<33
	8-257													
ENDSET	1-553#	2-7#												
ENDSFT	1-566#	2-7#	8-A78											
ENDSRV	1-578#	2-7#	6-269											
ENDSUB	1-594#	2-7#												
ENDSW	1-612#	2-7#	4-138											
ENDTST	1-622#	2-7#	7-23	7-49	7-74	7-99	7-141	8-8	8-41	8-79	8-115	8-150	8-182	8-213
	8-245	8-281	8-307	8-330	8-372	8-414	8-459	8-512	8-540	8-594	8-631	8-683	8-707	8-740
	8-770	8-805	8-821	8-846	8-894	8-937	8-:11	8-:35	8-:65	8-:11	8-<34	8-<82	8-=29	8-=76
	8->24	8-?23	8-258	8-A39										
EQUALS	1-640#	2-7#	2-26											
ERRDF	1-712#	2-7#	6-100	6-280	6-296	7-132	7-168	8-32	8-70	8-108	8-143	8-175	8-206	8-238
	8-279	8-304	8-327	8-359	8-368	8-402	8-412	8-446	8-455	8-490	8-499	8-507	8-572	8-581
	8-590	8-624	8-670	8-725	8-737	8-767	8-793	8-802	8-841	8-887	8-934	8-:04	8-:91	8-:08
	8-:82	8-<17	8-<79	8-=26	8-=73	8->21	8-?20	8-?87	8-223	8-242	8-A19			
ERRHRD	1-716#	2-7#												
ERROR	1-720#	2-7#												
ERRSF	1-724#	2-7#	7-21	7-47	7-72	7-97								
ERRSOF	1-728#	2-7#												
ERRTBL	1-732#	2-7#												
ESCAPE	1-742#	2-7#	7-133	7-169	8-33	8-71	8-109	8-144	8-176	8-207	8-239	8-566	8-659	8-662
	8-672	8-865	8-868	8-877	8-880	8-912	8-915	8-<08	8-<72	8-=18	8-=66	8->11	8->14	8-?13
EXIT	1-769#	2-7#												
FEQUAL	1-808#	2-7#												
GETBYT	1-822#	2-7#												
GETPRI	1-832#	2-7#												
GETWOR	1-827#	2-7#												
GMANIA	1-837#	2-7#												
GMANID	1-846#	2-7#												
GMANIL	1-857#	2-7#												
GPHARD	1-866#	2-7#	5-42											
GPRMA	1-872#	2-7#	8-A48	8-A50										
GPRMD	1-901#	2-7#	8-A46	8-A52	8-A56	8-A77								
GPRML	1-932#	2-7#	8-A54	8-A75										
HEADER	1-952#	2-7#	2-17											
INLOOP	1-960#	2-7#	6-7											
IOSETU	1-964#	2-7#												
IOSTAR	1-972#	2-7#												
KT11	1-980#	2-7#												
LASTAD	1-:45#	2-7#	8-A94											
MSBYTE	1-C98#	2-7#	2-17	2-17	2-17	2-17#								
MSCHEC	1-E16#	2-7#												
MSCNTO	1-E80#	2-7#	8-A46	8-A46#	8-A48	8-A48#	8-A50	8-A50#	8-A52	8-A52#	8-A54	8-A54#	8-A56	8-A56#
	8-A75	8-A75#	8-A77	8-A77#										
MSCOUN	1-D64#	2-7#	4-23	4-23	4-23#	4-32	4-32	4-32	4-32#	4-41	4-41	4-41#	4-60	4-60#
	4-67	4-67#	4-73	4-73	4-73#	4-76	4-76	4-76	4-76#	4-76	4-76	4-77	4-77	4-77
	4-77	4-77#	4-78	4-78	4-78	4-78	4-78	4-78	4-78#	4-79	4-79	4-79	4-79#	4-82

MSDATA	4-82# 1-B65# 2-17 2-17 2-21#	4-83 2-7# 2-17 2-17 2-22	4-83# 2-17 2-17 2-17 2-22#	5-86 2-17 2-17 2-17	5-86# 2-17 2-17 2-17	5-126 2-17 2-17 2-17	5-126# 2-17 2-17 2-17	5-137 2-17 2-17 2-17	5-137# 2-17 2-17 2-17	6-15 2-17 2-17 2-17	6-15# 2-17 2-17 2-17	2-17 2-17 2-17#	2-17 2-17 2-17#	2-17 2-17 2-21	
MSDECR	1-D27# 4-26 4-126 5-105 5-188 7-140 8-40# 8-115 8-212 8-307 8-594 8-770 8-937 8-<34 8-a57#	2-7# 4-26# 4-126# 5-105# 5-188# 7-140 8-40# 8-115# 8-212 8-307# 8-594# 8-770# 8-937# 8-<34# 8-a57#	2-19 2-19 4-128 5-141 6-269 7-140# 8-41 8-149 8-212# 8-330 8-631 8-805 8-:11 8-<82 8-a58	2-19# 4-35# 4-128# 5-141# 6-269# 7-140# 8-41# 8-149 8-212# 8-330# 8-631# 8-805# 8-:11# 8-<82# 8-a58#	2-72 4-44 4-138 5-164 6-304 7-141 8-78 8-149# 8-213 8-372 8-678 8-821 8-:35 8-=29 8-A39	2-72# 4-44# 4-138# 5-164# 6-304# 7-141# 8-78 8-149# 8-213# 8-372# 8-678# 8-821# 8-:35# 8-=29# 8-A39#	2-298 4-51 4-140 5-166 7-23 8-7 8-78# 8-150 8-244 8-414 8-678# 8-846 8-:65 8-=76 8-A58	2-298# 4-51# 4-140# 5-166# 7-23# 8-7 8-78# 8-150# 8-244 8-414# 8-678# 8-846# 8-:65# 8-=76# 8-A58#	3-91 4-62 4-146 5-176 7-49 8-7# 8-79 8-181 8-244# 8-459 8-683 8-893 8-:11 8->24 8-A68	3-91# 4-62# 4-146# 5-176# 7-49# 8-7# 8-79# 8-181 8-244# 8-459# 8-683# 8-893# 8-:11# 8->24# 8-A68#	4-11 4-71 5-7 5-178 7-74 8-8 8-114 8-181# 8-245 8-512 8-707 8-893# 8-<33 8->24# 8-A78	4-11# 4-71# 5-7# 5-178# 7-74# 8-8# 8-114 8-181# 8-245# 8-512# 8-707# 8-893# 8-<33 8->24# 8-A78#	4-18 4-114 5-103 5-186 7-99 8-40 8-114# 8-182 8-281 8-540 8-740 8-894 8-<33# 8-a57 8-A93	4-18# 4-114# 5-103# 5-186# 7-99# 8-40# 8-114# 8-182# 8-281# 8-540# 8-740# 8-894# 8-<33# 8-a57 8-A93#	4-18# 4-114# 5-103# 5-186# 7-99# 8-40# 8-114# 8-182# 8-281# 8-540# 8-740# 8-894# 8-<33# 8-a57 8-A93#
MSDEFA	1-E68# 8-A75	2-7# 8-A75#	8-A46 8-A77	8-A46# 8-A77#	8-A48 8-A77#	8-A48# 8-A50	8-A50# 8-A50	8-A52 8-A52	8-A52# 8-A52#	8-A54 8-A54	8-A54# 8-A54#	8-A56 8-A56	8-A56# 8-A56#	8-A56# 8-A56#	
MSENDE	1-D72# 4-114# 5-188# 8-79# 8-372# 8-846# 8-?23#	2-7# 4-126# 6-269# 8-114# 8-414# 8-893# 8-a57#	2-19# 4-128# 6-304# 8-115# 8-459# 8-894# 8-a58#	2-72# 4-138# 7-23# 8-149# 8-512# 8-937# 8-A39#	2-298# 4-140# 7-49# 8-150# 8-540# 8-:11# 8-A58#	3-91# 4-146# 7-74# 8-181# 8-594# 8-:35# 8-A68#	4-11# 5-103# 7-99# 8-182# 8-631# 8-:65# 8-A78#	4-18# 5-105# 7-140# 8-212# 8-678# 8-:11# 8-A93#	4-26# 5-141# 7-141# 8-213# 8-683# 8-<33# 8-A93#	4-35# 5-164# 8-7# 8-244# 8-707# 8-<34# 8-A93#	4-44# 5-166# 8-8# 8-245# 8-740# 8-<82# 8-A93#	4-51# 5-176# 8-40# 8-281# 8-770# 8-<82# 8-A93#	4-62# 5-178# 8-41# 8-307# 8-805# 8-<82# 8-A93#	4-71# 5-186# 8-78# 8-330# 8-821# 8->24# 8-A93#	
MSERRI	1-a47# 7-97 8-175 8-368 8-507 8-737 8-:04 8-=73	2-7# 7-97# 8-175# 8-368# 8-507# 8-737# 8-:04# 8-=73#	6-100 7-132 8-206 8-402 8-572 8-767 8-:91 8->21	6-100# 7-132# 8-206# 8-402# 8-572# 8-767# 8-:91# 8->21#	6-280 7-168 8-238 8-412 8-581 8-793 8-:08 8-?20	6-280# 7-168# 8-238# 8-412# 8-581# 8-793# 8-:08# 8-?20#	6-296 8-32 8-279 8-446 8-590 8-802 8-:82 8-?87	6-296# 8-32# 8-279# 8-446# 8-590# 8-802# 8-:82# 8-?87#	7-21 8-70 8-304 8-455 8-624 8-841 8-<17 8-a23	7-21# 8-70# 8-304# 8-455# 8-624# 8-841# 8-<17# 8-a23#	7-47 8-108 8-327 8-490 8-670 8-887 8-<79 8-a42	7-47# 8-108# 8-327# 8-490# 8-670# 8-887# 8-<79# 8-a42#	7-72 8-143 8-359 8-499 8-725 8-934 8-<26 8-A19	7-72# 8-143# 8-359# 8-499# 8-725# 8-934# 8-<26# 8-A19#	7-72# 8-143# 8-359# 8-499# 8-725# 8-934# 8-<26# 8-A19#
MSESCA	1-D04# 8-659# 8-<08#	2-7# 8-662 8-<72	7-133# 8-662# 8-<72#	7-169# 8-672# 8-=18	8-33# 8-865# 8-=18#	8-71# 8-865# 8-=66	8-109# 8-868# 8->11	8-144# 8-868# 8->11#	8-176# 8-877# 8->11#	8-207# 8-880# 8->14	8-239# 8-912 8->14#	8-566 8-912# 8-?13	8-566# 8-915 8-?13#	8-659 8-915# 8-144#	
MSESCS	1-D08# 8-176 8-877#	2-7# 8-176# 8-880	7-133 8-207 8-880#	7-133# 8-207# 8-912#	7-169 8-239 8-915#	8-33 8-566# 8-<08	8-33# 8-659# 8-<08#	8-71 8-662# 8-<72#	8-71# 8-672 8-=18#	8-109 8-672# 8->11#	8-109# 8-672# 8->14#	8-109# 8-865# 8-?13#	8-144# 8-868# 8-?13#	8-144# 8-877 8-A52#	
MSEXCP	1-D99# 8-A56	2-7# 8-A56	8-A46 8-A56#	8-A46 8-A77	8-A46# 8-A77	8-A48 8-A77#	8-A48# 8-A77#	8-A50 8-A77#	8-A50# 8-A77#	8-A50 8-A77#	8-A50# 8-A77#	8-A52 8-A77#	8-A52# 8-A77#	8-A52# 8-A77#	
MSEXIT	1-D12#	2-7#													
MSEXSE	1-D20#	2-7#													
MSEXTJ	1-D16#	2-7#													
MSGEN	1-D36# 2-17 2-17 2-17# 2-17# 2-17# 2-17# 2-24 4-18	2-7# 2-17 2-17 2-17# 2-17# 2-17# 2-17# 2-24# 4-18#	2-15 2-17 2-17 2-17# 2-17# 2-17# 2-17# 2-74 4-20	2-15# 2-17 2-17 2-17# 2-17# 2-17# 2-17# 2-74# 4-20#	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 3-1 4-26	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 3-1# 4-26#	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-3 4-28	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-3# 4-28#	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-5 4-35	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-5# 4-35#	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-11 4-37	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-11# 4-37#	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-13 4-44	2-17 2-17 2-17 2-17# 2-17# 2-17# 2-17# 4-13# 4-44#	

2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17
2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17	2-17
2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#
2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#
2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#	2-17#
2-22#	2-22#	4-11	4-11#	4-18	4-18#	4-23	4-23	4-23	4-23	4-23	4-23	4-23	4-23#
4-23#	4-23#	4-23#	4-23#	4-23#	4-26	4-26#	4-32	4-32	4-32	4-32	4-32	4-32	4-32#
4-32	4-32#	4-32#	4-32#	4-32#	4-32#	4-32#	4-35	4-35	4-35#	4-41	4-41	4-41	4-41
4-41	4-41	4-41	4-41#	4-41#	4-41#	4-41#	4-41#	4-41#	4-44	4-44#	4-51	4-51#	4-60
4-60	4-60	4-60	4-60	4-60#	4-60#	4-60#	4-62	4-62#	4-67	4-67	4-67	4-67	4-67
4-67	4-67	4-67#	4-67#	4-67#	4-67#	4-67#	4-71	4-71#	4-73	4-73	4-73	4-73	4-73
4-73	4-73	4-73	4-73#	4-73#	4-73#	4-73#	4-73#	4-73#	4-76	4-76	4-76	4-76	4-76
4-76	4-76	4-76	4-76	4-76	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#	4-76#
4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77	4-77#	4-77#	4-77#	4-77#	4-77#
4-77#	4-77#	4-77#	4-78	4-78	4-78	4-78	4-78	4-78	4-78	4-78	4-78	4-78	4-78#
4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#	4-78#
4-79	4-79	4-79	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#	4-79#
4-82	4-82	4-82	4-82#	4-82#	4-82#	4-82#	4-82#	4-82#	4-83	4-83	4-83	4-83	4-83
4-83#	4-83#	4-83#	4-83#	4-83#	4-83#	4-83#	4-83#	4-83#	4-83	4-83	4-83	4-83	4-83
4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-132	4-132#	4-144	4-144	4-144	4-144	4-144
4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144
4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144	4-144
4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#
4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#
4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#	4-144#
4-144#	5-14	5-14#	5-15	5-15	5-15#	5-15#	5-16	5-16#	5-19	5-19	5-19#	5-19#	5-20
5-20#	5-21	5-21	5-21#	5-21#	5-22	5-22#	5-30	5-30	5-30#	5-30#	5-31	5-31#	5-42
5-42	5-42#	5-42#	5-42#	5-43	5-43#	5-81	5-81	5-81	5-81	5-81	5-81	5-81	5-81
5-81#	5-86	5-86	5-86	5-86	5-86	5-86#	5-86#	5-86#	5-86#	5-89	5-89	5-89#	5-89#
5-90	5-90#	5-97	5-97	5-97	5-97	5-97	5-97	5-97#	5-97#	5-97#	5-97#	5-97#	5-97#
5-99	5-99#	5-100	5-100#	5-103	5-103#	5-112	5-112	5-112	5-112	5-112	5-112	5-112#	5-112#
5-112#	5-112#	5-112#	5-112#	5-121	5-121	5-121#	5-121#	5-126	5-126	5-126	5-126	5-126	5-126#
5-126#	5-126#	5-126#	5-128	5-128	5-128#	5-128#	5-137	5-137	5-137	5-137	5-137	5-137#	5-137#
5-137#	5-137#	5-139	5-139	5-139#	5-139#	5-141	5-141#	5-147	5-147	5-147#	5-147#	5-154	5-154
5-154#	5-154#	5-164	5-164#	5-176	5-176#	5-186	5-186#	6-7	6-7#	6-8	6-8#	6-15	6-15
6-15	6-15	6-15	6-15#	6-15#	6-15#	6-15#	6-17	6-17	6-17#	6-18	6-18#	6-100	6-100
6-100	6-100	6-100	6-100#	6-100#	6-100#	6-100#	6-100#	6-269	6-269#	6-276	6-276	6-276	6-276
6-276	6-276	6-276	6-276	6-276#	6-280	6-280	6-280	6-280	6-280#	6-280#	6-280#	6-280#	6-280#
6-290	6-290	6-290	6-290	6-290	6-290	6-290	6-290	6-290#	6-296	6-296	6-296	6-296	6-296#
6-296#	6-296#	6-296#	6-296#	7-13	7-13	7-13	7-13	7-13	7-13	7-13#	7-13#	7-13#	7-13#
7-13#	7-13#	7-16	7-16	7-16#	7-16#	7-21	7-21	7-21	7-21	7-21#	7-21#	7-21#	7-21#
7-21#	7-22	7-22#	7-23	7-23#	7-39	7-39	7-39	7-39	7-39	7-39#	7-39#	7-39#	7-39#
7-39#	7-39#	7-39#	7-42	7-42	7-42#	7-42#	7-47	7-47	7-47	7-47#	7-47#	7-47#	7-47#
7-47#	7-47#	7-48	7-48#	7-49	7-49#	7-64	7-64	7-64	7-64	7-64	7-64	7-64#	7-64#
7-64#	7-64#	7-64#	7-64#	7-67	7-67	7-67#	7-72	7-72	7-72	7-72	7-72	7-72#	7-72#
7-72#	7-72#	7-72#	7-73	7-73#	7-74	7-74#	7-89	7-89	7-89	7-89	7-89	7-89	7-89#
7-89#	7-89#	7-89#	7-89#	7-92	7-92	7-92#	7-92#	7-92#	7-97	7-97	7-97	7-97	7-97#
7-97#	7-97#	7-97#	7-97#	7-98	7-98#	7-99	7-99#	7-99#	7-118	7-118#	7-132	7-132	7-132
7-132#	7-132#	7-132#	7-132#	7-133	7-133	7-133	7-133#	7-133#	7-133#	7-140	7-141	7-141#	7-157
7-157#	7-168	7-168	7-168	7-168	7-168#	7-168#	7-168#	7-168#	7-168#	7-169	7-169	7-169#	7-169#
8-7	8-7#	8-8	8-8#	8-23	8-23#	8-32	8-32	8-32	8-32	8-32	8-32#	8-32#	8-32#
8-32#	8-33	8-33	8-33#	8-33#	8-40	8-40#	8-41	8-41#	8-56	8-56#	8-70	8-70	8-70
8-70	8-70#	8-70#	8-70#	8-70#	8-70#	8-71	8-71	8-71#	8-71#	8-78	8-78#	8-79	8-79#
8-94	8-94#	8-108	8-108	8-108	8-108	8-108#	8-108#	8-108#	8-108#	8-108#	8-109	8-109	8-109#
8-109#	8-114	8-114#	8-115	8-115#	8-131	8-131#	8-143	8-143	8-143	8-143	8-143#	8-143#	8-143#

8-143#	8-143#	8-144	8-144	8-144#	8-144#	8-149	8-149#	8-150	8-150#	8-165	8-165#	8-175	8-175
8-175	8-175	8-175#	8-175#	8-175#	8-175#	8-175#	8-176	8-176	8-176#	8-176#	8-181	8-181#	8-182
8-182#	8-197	8-197#	8-206	8-206	8-206	8-206	8-206#	8-206#	8-206#	8-206#	8-206#	8-207	8-207
8-207#	8-207#	8-212	8-212#	8-213	8-213#	8-228	8-228#	8-238	8-238	8-238	8-238	8-238#	8-238#
8-238#	8-238#	8-238#	8-239	8-239	8-239#	8-239#	8-244	8-244#	8-245	8-245#	8-264	8-264	8-264#
8-264#	8-271	8-271#	8-279	8-279	8-279	8-279	8-279#	8-279#	8-279#	8-279#	8-279#	8-281	8-281#
8-300	8-300#	8-304	8-304	8-304	8-304	8-304#	8-304#	8-304#	8-304#	8-304#	8-307	8-307#	8-323
8-323#	8-327	8-327	8-327	8-327	8-327#	8-327#	8-327#	8-327#	8-327#	8-330	8-330#	8-359	8-359
8-359	8-359	8-359#	8-359#	8-359#	8-359#	8-359#	8-360	8-360#	8-368	8-368	8-368	8-368	8-368#
8-368#	8-368#	8-368#	8-368#	8-372	8-372#	8-402	8-402	8-402	8-402	8-402#	8-402#	8-402#	8-402#
8-402#	8-403	8-403#	8-412	8-412	8-412	8-412	8-412#	8-412#	8-412#	8-412#	8-412#	8-414	8-414#
8-446	8-446	8-446	8-446	8-446#	8-446#	8-446#	8-446#	8-446#	8-447	8-447#	8-455	8-455	8-455
8-455	8-455#	8-455#	8-455#	8-455#	8-455#	8-459	8-459#	8-490	8-490	8-490	8-490	8-490#	8-490#
8-490#	8-490#	8-490#	8-491	8-491#	8-499	8-499	8-499	8-499	8-499#	8-499#	8-499#	8-499#	8-499#
8-500	8-500#	8-507	8-507	8-507	8-507	8-507#	8-507#	8-507#	8-507#	8-507#	8-512	8-512#	8-535
8-535#	8-540	8-540#	8-563	8-563#	8-566	8-566	8-566#	8-566#	8-572	8-572	8-572	8-572#	8-572#
8-572#	8-572#	8-572#	8-572#	8-574	8-574#	8-581	8-581	8-581	8-581	8-581#	8-581#	8-581#	8-581#
8-581#	8-583	8-583#	8-590	8-590	8-590	8-590	8-590#	8-590#	8-590#	8-590#	8-590#	8-594	8-594#
8-618	8-618	8-618#	8-618#	8-624	8-624	8-624	8-624#	8-624#	8-624#	8-624#	8-624#	8-624#	8-626
8-626#	8-631	8-631#	8-650	8-650#	8-653	8-653	8-653#	8-653#	8-659	8-659	8-659#	8-659#	8-662
8-662	8-662#	8-662#	8-670	8-670	8-670	8-670	8-670#	8-670#	8-670#	8-670#	8-670#	8-672	8-672
8-672#	8-672#	8-678	8-678#	8-683	8-683#	8-703	8-703#	8-707	8-707#	8-717	8-717	8-717#	8-717#
8-722	8-722	8-722#	8-722#	8-725	8-725	8-725	8-725	8-725#	8-725#	8-725#	8-725#	8-725#	8-726
8-726#	8-729	8-729	8-729#	8-729#	8-734	8-734	8-734#	8-734#	8-737	8-737	8-737	8-737	8-737#
8-737#	8-737#	8-737#	8-737#	8-738	8-738#	8-740	8-740#	8-764	8-764#	8-767	8-767	8-767	8-767
8-767#	8-767#	8-767#	8-767#	8-767#	8-770	8-770#	8-784	8-784	8-784#	8-784#	8-790	8-790	8-790#
8-790#	8-793	8-793	8-793	8-793	8-793#	8-793#	8-793#	8-793#	8-793#	8-794	8-794#	8-799	8-799#
8-802	8-802	8-802	8-802	8-802#	8-802#	8-802#	8-802#	8-802#	8-805	8-805#	8-818	8-818#	8-821
8-821#	8-833	8-833	8-833#	8-833#	8-838	8-838	8-838#	8-838#	8-841	8-841	8-841	8-841	8-841#
8-841#	8-841#	8-841#	8-841#	8-842	8-842#	8-846	8-846#	8-865	8-865	8-865#	8-865#	8-868	8-868
8-868#	8-868#	8-872	8-872#	8-877	8-877	8-877#	8-877#	8-880	8-880	8-880#	8-880#	8-887	8-887
8-887	8-887	8-887#	8-887#	8-887#	8-887#	8-887#	8-889	8-889#	8-893	8-893#	8-894	8-894#	8-912
8-912	8-912#	8-912#	8-915	8-915	8-915#	8-915#	8-934	8-934	8-934	8-934	8-934#	8-934#	8-934#
8-934#	8-934#	8-937	8-937#	8-:04	8-:04	8-:04	8-:04	8-:04#	8-:04#	8-:04#	8-:04#	8-:04#	8-:05
8-:05#	8-:11	8-:11#	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:29	8-:32	8-:32#
8-:35	8-:35#	8-:55	8-:55#	8-:58	8-:58#	8-:61	8-:61#	8-:65	8-:65#	8-:83	8-:83	8-:83#	8-:83#
8-:91	8-:91	8-:91	8-:91	8-:91#	8-:91#	8-:91#	8-:91#	8-:91#	8-:92	8-:92#	8-:96	8-:96#	8-:96#
8-:02#	8-:04	8-:04	8-:04#	8-:04#	8-:08	8-:08	8-:08	8-:08	8-:08#	8-:08#	8-:08#	8-:08#	8-:08#
8-:09	8-:09#	8-:11	8-:11#	8-:34	8-:34#	8-:39	8-:39#	8-:42	8-:42#	8-:57	8-:57#	8-:60	8-:60#
8-:63	8-:63#	8-:66	8-:66#	8-:71	8-:71#	8-:74	8-:74#	8-:82	8-:82	8-:82	8-:82	8-:82#	8-:82#
8-:82#	8-:82#	8-:82#	8-:83	8-:83#	8-:90	8-:90#	8-:93	8-:93#	8-:96	8-:96#	8-:99	8-:99#	8-:05
8-<05#	8-<08	8-<08	8-<08#	8-<08#	8-<17	8-<17	8-<17	8-<17	8-<17#	8-<17#	8-<17#	8-<17#	8-<17#
8-<18	8-<18#	8-<33	8-<33#	8-<34	8-<34#	8-<55	8-<55#	8-<58	8-<58#	8-<61	8-<61#	8-<64	8-<64#
8-<69	8-<69#	8-<72	8-<72#	8-<72#	8-<72#	8-<79	8-<79	8-<79	8-<79	8-<79#	8-<79#	8-<79#	8-<79#
8-<79#	8-<82	8-<82#	8- =01	8- =01#	8- =04	8- =04#	8- =07	8- =07#	8- =10	8- =10#	8- =15	8- =15#	8- =18
8- =18	8- =18#	8- =18#	8- =26	8- =26	8- =26	8- =26	8- =26#	8- =26#	8- =26#	8- =26#	8- =26#	8- =29	8- =29#
8- =48	8- =48#	8- =51	8- =51#	8- =54	8- =54#	8- =57	8- =57#	8- =63	8- =63#	8- =66	8- =66#	8- =66#	8- =66#
8- =73	8- =73	8- =73	8- =73#	8- =73#	8- =73#	8- =73#	8- =73#	8- =73#	8- =76	8- =76#	8- =96	8- =96#	8- =99
8- =99#	8- >02	8- >02#	8- >05	8- >05#	8- >11	8- >11	8- >11#	8- >11#	8- >14	8- >14	8- >14#	8- >14#	8- >21
8- >21	8- >21	8- >21	8- >21#	8- >21#	8- >21#	8- >21#	8- >21#	8- >24	8- >24#	8- >46	8- >46#	8- >49	8- >49#
8- >87	8- >87#	8- >91	8- >91#	8- >95	8- >95#	8- >98	8- >98#	8- ?01	8- ?01#	8- ?05	8- ?05#	8- ?10	8- ?10#
8- ?13	8- ?13	8- ?13#	8- ?13#	8- ?20	8- ?20	8- ?20	8- ?20	8- ?20#	8- ?20#	8- ?20#	8- ?20#	8- ?20#	8- ?23
8- ?23#	8- ?38	8- ?38#	8- ?43	8- ?43#	8- ?46	8- ?46#	8- ?61	8- ?61#	8- ?64	8- ?64#	8- ?67	8- ?67#	8- ?70
8- ?70#	8- ?76	8- ?76#	8- ?79	8- ?79#	8- ?87	8- ?87	8- ?87	8- ?87	8- ?87#	8- ?87#	8- ?87#	8- ?87#	8- ?87#
8- ?88	8- ?88#	8- ?95	8- ?95#	8- ?98	8- ?98#	8- @01	8- @01#	8- @05	8- @05#	8- @10	8- @10#	8- @14	8- @14#
8- @23	8- @23	8- @23	8- @23	8- @23#	8- @23#	8- @23#	8- @23#	8- @23#	8- @24	8- @24#	8- @42	8- @42	8- @42

	8-359#	8-360#	8-368#	8-372#	8-377	8-377	8-377	8-377#	8-377#	8-377#	8-402#	8-403#	8-412#	8-414#
	8-419	8-419	8-419	8-419#	8-419#	8-419#	8-446#	8-447#	8-455#	8-459#	8-463	8-463	8-463	8-463#
	8-463#	8-463#	8-490#	8-491#	8-499#	8-500#	8-507#	8-512#	8-516	8-516	8-516	8-516#	8-516#	8-516#
	8-535#	8-540#	8-545	8-545	8-545	8-545#	8-545#	8-545#	8-563#	8-566#	8-572#	8-574#	8-581#	8-583#
	8-590#	8-594#	8-599	8-599	8-599	8-599#	8-599#	8-599#	8-618#	8-624#	8-626#	8-631#	8-636	8-636
	8-636	8-636#	8-636#	8-636#	8-650	8-650	8-650	8-650#	8-650#	8-650#	8-650#	8-653#	8-659#	8-662#
	8-670#	8-672#	8-678#	8-683#	8-687	8-687	8-687	8-687#	8-687#	8-687#	8-703#	8-707#	8-712	8-712
	8-712	8-712#	8-712#	8-712#	8-717#	8-722#	8-725#	8-726#	8-729#	8-734#	8-737#	8-738#	8-740#	8-745
	8-745	8-745	8-745#	8-745#	8-745#	8-764#	8-767#	8-770#	8-775	8-775	8-775	8-775#	8-775#	8-775#
	8-784#	8-790#	8-793#	8-794#	8-799#	8-802#	8-805#	8-809	8-809	8-809	8-809#	8-809#	8-809#	8-818#
	8-821#	8-825	8-825	8-825	8-825#	8-825#	8-825#	8-833#	8-838#	8-841#	8-842#	8-846#	8-851	8-851
	8-851	8-851#	8-851#	8-851#	8-865#	8-868#	8-872	8-872	8-872	8-872#	8-872#	8-872#	8-872#	8-877#
	8-880#	8-887#	8-889#	8-893#	8-894#	8-899	8-899	8-899	8-899#	8-899#	8-899#	8-912#	8-915#	8-934#
	8-937#	8-942	8-942	8-942	8-942#	8-942#	8-942#	8-942#	8-942#	8-942#	8-942#	8-942#	8-942#	8-942#
	8-:16#	8-:16#	8-:32#	8-:35#	8-:40	8-:40	8-:40	8-:40#	8-:40#	8-:40#	8-:40#	8-:40#	8-:40#	8-:40#
	8-:70	8-:70	8-:70	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#	8-:70#
	8-:11#	8-:16	8-:16	8-:16	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#	8-:16#
	8-:39#	8-:42#	8-:57#	8-:60#	8-:63#	8-:66#	8-:71#	8-:74#	8-:82#	8-:83#	8-:90#	8-:93#	8-:96#	8-:99#
	8-<05#	8-<08#	8-<17#	8-<18#	8-<33#	8-<34#	8-<39	8-<39	8-<39	8-<39#	8-<39#	8-<39#	8-<39#	8-<39#
	8-<61#	8-<64#	8-<69#	8-<72#	8-<79#	8-<82#	8-<87	8-<87	8-<87	8-<87#	8-<87#	8-<87#	8-<87#	8-<87#
	8-=07#	8-=10#	8-=15#	8-=18#	8-=26#	8-=29#	8-=34	8-=34	8-=34	8-=34#	8-=34#	8-=34#	8-=34#	8-=34#
	8-=54#	8-=57#	8-=63#	8-=66#	8-=73#	8-=76#	8-=81	8-=81	8-=81	8-=81#	8-=81#	8-=81#	8-=81#	8-=81#
	8->02#	8->05#	8->11#	8->14#	8->21#	8->24#	8->29	8->29	8->29	8->29#	8->29#	8->29#	8->29#	8->29#
	8->87#	8->91#	8->95#	8->98#	8-?01#	8-?05#	8-?10#	8-?13#	8-?20#	8-?23#	8-?28	8-?28	8-?28	8-?28#
	8-?28#	8-?28#	8-?38	8-?38	8-?38	8-?38#	8-?38#	8-?38#	8-?38#	8-?38#	8-?43#	8-?46#	8-?61#	8-?64#
	8-?70#	8-?76#	8-?79#	8-?87#	8-?88#	8-?95#	8-?98#	8-?01#	8-?05#	8-?10#	8-?13#	8-?14#	8-?23#	8-?24#
	8-a43#	8-a57#	8-a58#	8-a63	8-a63	8-a63	8-a63#	8-a63#	8-a63#	8-a63#	8-a75#	8-a78#	8-a93#	8-a99#
	8-A02#	8-A08#	8-A11#	8-A19#	8-A20#	8-A27#	8-A31#	8-A36#	8-A39#	8-A41	8-A41#	8-A43	8-A43	8-A43#
	8-A43#	8-A72	8-A72#	8-A74	8-A74	8-A74#	8-A74#	8-A74#	8-A74#	8-A74#	8-A74#	8-A74#	8-A74#	8-A74#
MSIOSE	1-a98#	2-7#												
MSLDRO	1-c40#	2-7#	5-15	5-15#	5-19	5-19#	5-21	5-21#	5-30	5-30#	5-42	5-42#	5-89	5-89#
	5-121	5-121#	5-128	5-128#	5-139	5-139#	5-147	5-147#	5-154	5-154#	6-17	6-17#	7-16	7-16#
	7-42	7-42#	7-67	7-67#	7-92	7-92#	8-264	8-264#	8-618	8-618#	8-653	8-653#	8-717	8-717#
	8-722	8-722#	8-729	8-729#	8-734	8-734#	8-784	8-784#	8-790	8-790#	8-833	8-833#	8-838	8-838#
	8-:83	8-:83#	8-:04	8-:04#										
MSMASK	1-a69#	2-7#												
MSMCHI	1-2#	2-7	2-7#	2-7#										
MSMCLO	1-a22#	2-7	2-7#	2-7#										
MSMSK1	1-a75#	2-7#												
MSPOP	1-B79#	2-7#	2-19	2-19#	2-72	2-72#	2-298	2-298#	3-91	3-91#	4-11	4-11#	4-18	4-18#
	4-26	4-26#	4-35	4-35#	4-44	4-44#	4-51	4-51#	4-62	4-62#	4-71	4-71#	4-114	4-114#
	4-126	4-126#	4-128	4-128#	4-138	4-138#	4-140	4-140#	4-146	4-146#	5-7	5-7#	5-103	5-103#
	5-105	5-105#	5-141	5-141#	5-164	5-164#	5-166	5-166#	5-176	5-176#	5-178	5-178#	5-186	5-186#
	5-188	5-188#	6-269	6-269#	6-304	6-304#	7-23	7-23#	7-49	7-49#	7-74	7-74#	7-99	7-99#
	7-140	7-140	7-140#	7-141	7-141#	8-7	8-7	8-7#	8-8	8-8#	8-40	8-40	8-40#	8-41
	8-41#	8-78	8-78	8-78#	8-79	8-79#	8-114	8-114	8-114#	8-115	8-115#	8-149	8-149	8-149#
	8-150	8-150#	8-181	8-181	8-181#	8-182	8-182#	8-212	8-212	8-212#	8-213	8-213#	8-244	8-244
	8-244#	8-245	8-245#	8-281	8-281#	8-307	8-307#	8-330	8-330#	8-372	8-372#	8-414	8-414#	8-459
	8-459#	8-512	8-512#	8-540	8-540#	8-594	8-594#	8-631	8-631#	8-678	8-678	8-678#	8-683	8-683#
	8-707	8-707#	8-740	8-740#	8-770	8-770#	8-805	8-805#	8-821	8-821#	8-846	8-846#	8-893	8-893#
	8-893#	8-894	8-894#	8-937	8-937#	8-:11	8-:11#	8-:35	8-:35#	8-:65	8-:65#	8-:11	8-:11#	8-<33
	8-<33	8-<33#	8-<34	8-<34#	8-<82	8-<82#	8-=29	8-=29#	8-=76	8-=76#	8->24	8->24#	8-?23	8-?23#
	8-a57	8-a57	8-a57#	8-a58	8-a58#	8-A39	8-A39#	8-A58	8-A58#	8-A68	8-A68#	8-A78	8-A78#	8-A93
	8-A93#													
MSPRIN	1-a34#	2-7#	4-23	4-23#	4-32	4-32#	4-41	4-41#	4-60	4-60#	4-67	4-67#	4-73	4-73#
	4-76	4-76#	4-77	4-77#	4-78	4-78#	4-79	4-79#	4-82	4-82#	4-83	4-83#	5-86	5-86#

MSPUSH	5-126 1-229# 4-13 4-116 5-12 5-182 7-104 8-23# 8-131 8-228 8-419# 8-687 8-872 8-:16# 8-:34 8-?28 8-A74#	5-126# 2-7# 4-13# 4-116# 5-12# 5-182# 7-104# 8-46 8-131# 8-228 8-463 8-687# 8-872 8-:34 8-?28#	5-137 2-15 4-20 4-118 5-108 6-4 7-118 8-46# 8-155 8-228# 8-463# 8-712 8-872# 8-:34 8-?38	5-137# 2-15# 4-20# 4-118# 5-108# 6-4# 7-118 8-56 8-155# 8-250 8-516 8-712# 8-899 8-:34# 8-?38	6-15 2-24 4-28 4-130 5-143 6-265 7-118# 8-56 8-165 8-250# 8-516# 8-745 8-899# 8-<39 8-?38#	6-15# 2-24# 4-28# 4-130# 5-143# 6-265# 7-146 8-56# 8-165 8-286 8-545 8-745# 8-942 8-<39# 8-263	2-74 4-37 4-132 5-145 7-3 7-146# 8-84 8-165# 8-286# 8-545# 8-775 8-942# 8-<87 8-263#	2-74# 4-37# 4-132# 5-145# 7-3# 7-157 8-84# 8-187 8-312 8-599 8-775# 8-942# 8-<87# 8-A41	3-1 4-46 4-142 5-170 7-28 7-157 8-94 8-187# 8-312# 8-599# 8-809 8-:16# 8-:34 8-A41#	3-1# 4-46# 4-142# 5-170# 7-28# 7-157# 8-94 8-197 8-335 8-636 8-809# 8-:40 8-:34# 8-A43	4-3 4-53 5-3 5-172 7-54 8-13 8-94# 8-197 8-335# 8-636# 8-825 8-:40# 8-:81 8-A43#	4-3# 4-53# 5-3# 5-172# 7-54# 8-13# 8-120 8-197# 8-377 8-650 8-825# 8-:70 8-:81# 8-A72	4-5 4-64 5-10 5-180 7-79 8-23 8-120# 8-218 8-377# 8-650 8-851 8-:70# 8->29 8-A72#	4-5# 4-64# 5-10# 5-180# 7-79# 8-23 8-131 8-218# 8-419 8-650# 8-851# 8-:16 8->29# 8-A74
MSPUT	1-C70# 4-41 4-73 4-77 4-79 5-86# 5-137 7-39 1-C79# 4-32 4-60 4-73# 4-76# 4-77# 4-78# 4-82 5-86# 5-112# 6-15# 7-39# 7-89	2-7# 4-41 4-73# 4-77 4-79 5-97 5-137 7-39# 2-7# 4-32# 4-60 4-73# 4-76# 4-77# 4-78# 4-82 5-86# 5-112# 6-15# 7-39# 7-89	4-23 4-41 4-76 4-77# 4-79 5-97 5-137# 7-64 4-23 4-32# 4-60# 4-73# 4-76# 4-77# 4-78 4-82 5-97 5-112# 7-13 7-39# 7-89#	4-23 4-41# 4-76 4-78 4-79# 5-97 6-15 7-64 4-23 4-32# 4-60# 4-73# 4-76# 4-77 4-78 4-82# 5-97 5-112# 7-13 7-39# 7-89#	4-23 4-60 4-76 4-78 4-82 5-97 6-15 7-64 4-23 4-32# 4-67 4-76 4-77 4-78 4-82# 5-97 5-126 7-13 7-64 7-89#	4-23 4-60 4-76 4-78 4-82 5-97# 6-15# 7-64 4-23# 4-32# 4-67 4-76 4-77 4-78 4-82# 5-97 5-126 7-13 7-64 7-89#	4-23# 4-60# 4-76 4-78 4-82 5-112 7-13 7-64# 4-23# 4-41 4-67 4-76 4-77 4-78 4-83 5-97# 5-126# 7-13# 7-64 7-89#	4-32 4-67 4-76 4-78 4-82# 5-112 7-13 7-89 4-23# 4-41 4-67# 4-76 4-77 4-78 4-83 5-97# 5-126# 7-13# 7-64 7-89#	4-32 4-67 4-76 4-78 4-83 5-112 7-13 7-89 4-23# 4-41 4-67# 4-76 4-77 4-78 4-83 5-97# 5-137 7-13# 7-64# 7-89#	4-32 4-67# 4-77 4-78# 4-83 5-112# 7-13# 7-89 4-32 4-41# 4-67# 4-76 4-77# 4-78# 4-83# 5-112 5-137# 7-39 7-64# 7-89#	4-32 4-73 4-77 4-79 4-83# 5-126 7-39 7-89# 4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 5-137# 7-39 7-64# 7-89#	4-32# 4-73 4-77 4-79 5-86 5-126 7-39 4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 6-15 7-39 7-89#	4-41 4-73 4-77 4-79 5-86 5-126# 7-39 4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 6-15 7-39 7-89#	
MSPUT1	1-C79# 4-32 4-60 4-73# 4-76# 4-77# 4-78# 4-82 5-86# 5-112# 6-15# 7-39# 7-89	2-7# 4-32# 4-60 4-73# 4-76# 4-77# 4-78# 4-82 5-86# 5-112# 6-15# 7-39# 7-89	4-23 4-32# 4-60# 4-73# 4-76# 4-77# 4-78 4-82 5-97 5-112# 7-13 7-39# 7-89#	4-23 4-32# 4-60# 4-73# 4-76# 4-77 4-78 4-82# 5-97 5-112# 7-13 7-39# 7-89#	4-23 4-32# 4-67 4-76 4-77 4-78 4-82# 5-97 5-126 7-13 7-64 7-89#	4-23# 4-32# 4-67 4-76 4-77 4-78 4-82# 5-97 5-126 7-13# 7-64 7-89#	4-23# 4-41 4-67 4-76 4-77 4-78 4-83 5-97# 5-126# 7-13# 7-64 7-89#	4-23# 4-41 4-67# 4-76 4-77 4-78 4-83 5-97# 5-137 7-13# 7-64# 7-89#	4-23# 4-41 4-67# 4-76 4-77 4-78 4-83 5-97# 5-137 7-13# 7-64# 7-89#	4-32 4-41# 4-67# 4-76 4-77# 4-78# 4-83# 5-112 5-137# 7-39 7-64# 7-89#	4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 5-137# 7-39 7-64# 7-89#	4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 6-15 7-39 7-89#	4-32 4-41# 4-73 4-76# 4-77# 4-78# 4-83# 5-112 6-15 7-39 7-89#	
MSRADI	1-D75# 8-A75	2-7# 8-A75#	8-A46 8-A77	8-A46# 8-A77#	8-A48 8-A48#	8-A50 8-A50#	8-A52 8-A52#	8-A54 8-A54#	8-A56 8-A56#	8-A56 8-A56#	8-A56 8-A56#	8-A56 8-A56#	8-A56# 8-A56#	
MSRBRO	1-C50#	2-7#												
MSRNRO	1-C60#	2-7#												
MSSETS	1-D30# 4-13 4-116 5-12 5-182 7-104 8-23 8-94# 8-187 8-286 8-545 8-745 8-899 8-:34# 8-?38	2-7# 4-13# 4-116# 5-12# 5-182# 7-104# 8-23 8-94# 8-187# 8-286# 8-545# 8-745# 8-899# 8-:34# 8-?38	5-42 2-15 4-20 4-118 5-108 6-4 7-118 8-23# 8-120 8-197 8-312 8-599 8-775 8-942 8-<39 8-?38#	5-42# 2-15# 4-20# 4-118# 5-108# 6-4# 7-118 8-23# 8-120# 8-197# 8-312# 8-599# 8-775# 8-942# 8-<39# 8-?38#	2-24 4-28 4-130 5-143 6-265 7-118# 8-46 8-131 8-197# 8-335 8-636 8-809 8-:16 8-<87 8-263	2-24# 4-28# 4-130# 5-143# 6-265# 7-118# 8-46# 8-131 8-197# 8-335# 8-636# 8-809# 8-:16# 8-<87# 8-263#	2-74 4-37 4-132 5-145 7-3 7-146 8-56 8-131# 8-218 8-377 8-650 8-825 8-:40 8-:34# 8-A41	2-74# 4-37# 4-132# 5-145# 7-3# 7-146# 8-56 8-131# 8-218# 8-377# 8-650# 8-825# 8-:40# 8-:34# 8-A41#	3-1 4-46 4-142 5-170 7-28 7-157 8-56# 8-155 8-228 8-419 8-650# 8-851 8-:70 8-:81 8-A43	3-1# 4-46# 4-142# 5-170# 7-28# 7-157# 8-56# 8-155# 8-228 8-419# 8-650# 8-851# 8-:70# 8-:81# 8-A43	4-3 4-53 5-3 5-172 7-54 7-157# 8-84 8-165 8-228# 8-463 8-687 8-872 8-:16# 8-:81# 8-A72	4-3# 4-53# 5-3# 5-172# 7-54# 7-157# 8-84# 8-165 8-228# 8-463# 8-687# 8-872# 8-:16# 8->29# 8-A72#	4-5 4-64 5-10 5-180 7-79 8-13 8-94 8-165# 8-250 8-516 8-712 8-872# 8-:34 8-?28 8-A74	4-5# 4-64# 5-10# 5-180# 7-79# 8-13# 8-94# 8-165# 8-250# 8-516# 8-712# 8-872# 8-:34 8->29# 8-A74

MSTSTL

8-703#	8-707#	8-717#	8-722#	8-725#	8-726#	8-729#	8-734#	8-737#	8-738#	8-740#	8-764#	8-767#	8-770#
8-784#	8-790#	8-793#	8-794#	8-799#	8-802#	8-805#	8-818#	8-821#	8-833#	8-838#	8-841#	8-842#	8-846#
8-865#	8-868#	8-872#	8-877#	8-880#	8-887#	8-889#	8-893#	8-894#	8-912#	8-915#	8-934#	8-937#	8-:04#
8-:05#	8-:11#	8-:32#	8-:35#	8-:55#	8-:58#	8-:61#	8-:65#	8-:83#	8-:91#	8-:92#	8-:96#	8-:02#	8-:04#
8-:08#	8-:09#	8-:11#	8-:34#	8-:39#	8-:42#	8-:57#	8-:60#	8-:63#	8-:66#	8-:71#	8-:74#	8-:82#	8-:83#
8-:90#	8-:93#	8-:96#	8-:99#	8-<05#	8-<08#	8-<17#	8-<18#	8-<33#	8-<34#	8-<55#	8-<58#	8-<61#	8-<64#
8-<69#	8-<72#	8-<79#	8-<82#	8-=01#	8-=04#	8-=07#	8-=10#	8-=15#	8-=18#	8-=26#	8-=29#	8-=48#	8-=51#
8-=54#	8-=57#	8-=63#	8-=66#	8-=73#	8-=76#	8-=96#	8-=99#	8->02#	8->05#	8->11#	8->14#	8->21#	8->24#
8->46#	8->49#	8->87#	8->91#	8->95#	8->98#	8-?01#	8-?05#	8-?10#	8-?13#	8-?20#	8-?23#	8-?38#	8-?43#
8-?46#	8-?61#	8-?64#	8-?67#	8-?70#	8-?76#	8-?79#	8-?87#	8-?88#	8-?95#	8-?98#	8-a01#	8-a05#	8-a10#
8-a14#	8-a23#	8-a24#	8-a42#	8-a43#	8-a57#	8-a58#	8-a75#	8-a78#	8-a93#	8-a96#	8-a99#	8-A02#	8-A08#
8-A11#	8-A19#	8-A20#	8-A27#	8-A31#	8-A36#	8-A39#							
1-C19#	2-7#	4-11	4-11#	4-18	4-18#	4-23	4-23#	4-26	4-26#	4-32	4-32#	4-35	4-35#
4-41	4-41#	4-44	4-44#	4-51	4-51#	4-60	4-60#	4-62	4-62#	4-67	4-67#	4-71	4-71#
4-73	4-73#	4-76	4-76#	4-77	4-77#	4-78	4-78#	4-79	4-79#	4-82	4-82#	4-83	4-83#
5-14	5-14#	5-15	5-15#	5-19	5-19#	5-21	5-21#	5-30	5-30#	5-42	5-42#	5-86	5-86#
5-89	5-89#	5-90	5-90#	5-97	5-97#	5-99	5-99#	5-103	5-103#	5-112	5-112#	5-121	5-121#
5-126	5-126#	5-128	5-128#	5-137	5-137#	5-139	5-139#	5-141	5-141#	5-147	5-147#	5-154	5-154#
5-164	5-164#	5-176	5-176#	5-186	5-186#	6-7	6-7#	6-15	6-15#	6-17	6-17#	6-18	6-18#
6-100	6-100#	6-100#	6-280	6-280#	6-280#	6-296	6-296#	6-296#	7-13	7-13#	7-16	7-16#	7-21
7-21#	7-21#	7-22	7-22#	7-23	7-23#	7-39	7-39#	7-42	7-42#	7-47	7-47#	7-47#	7-48
7-48#	7-49	7-49#	7-64	7-64#	7-67	7-67#	7-72	7-72#	7-72#	7-73	7-73#	7-74	7-74#
7-89	7-89#	7-92	7-92#	7-97	7-97#	7-97#	7-98	7-98#	7-99	7-99#	7-118	7-118#	7-132
7-132#	7-132#	7-133	7-133#	7-140	7-140#	7-141	7-141#	7-157	7-157#	7-168	7-168#	7-168#	7-169
7-169#	8-7	8-7#	8-8	8-8#	8-23	8-23#	8-32	8-32#	8-32#	8-33	8-33#	8-40	8-40#
8-41	8-41#	8-56	8-56#	8-70	8-70#	8-70#	8-71	8-71#	8-78	8-78#	8-79	8-79#	8-94
8-94#	8-108	8-108#	8-108#	8-109	8-109#	8-114	8-114#	8-115	8-115#	8-131	8-131#	8-143	8-143#
8-143#	8-144	8-144#	8-149	8-149#	8-150	8-150#	8-165	8-165#	8-175	8-175#	8-175#	8-176	8-176#
8-181	8-181#	8-182	8-182#	8-197	8-197#	8-206	8-206#	8-206#	8-207	8-207#	8-212	8-212#	8-213
8-213#	8-228	8-228#	8-238	8-238#	8-238#	8-239	8-239#	8-244	8-244#	8-245	8-245#	8-264	8-264#
8-271	8-271#	8-279	8-279#	8-279#	8-281	8-281#	8-300	8-300#	8-304	8-304#	8-304#	8-307	8-307#
8-323	8-323#	8-327	8-327#	8-327#	8-330	8-330#	8-359	8-359#	8-359#	8-360	8-360#	8-368	8-368#
8-368#	8-372	8-372#	8-402	8-402#	8-402#	8-403	8-403#	8-412	8-412#	8-412#	8-414	8-414#	8-446
8-446#	8-446#	8-447	8-447#	8-455	8-455#	8-455#	8-459	8-459#	8-490	8-490#	8-490#	8-491	8-491#
8-499	8-499#	8-499#	8-500	8-500#	8-507	8-507#	8-507#	8-512	8-512#	8-535	8-535#	8-540	8-540#
8-563	8-563#	8-566	8-566#	8-572	8-572#	8-572#	8-574	8-574#	8-581	8-581#	8-581#	8-583	8-583#
8-590	8-590#	8-590#	8-594	8-594#	8-618	8-618#	8-624	8-624#	8-624#	8-626	8-626#	8-631	8-631#
8-650	8-650#	8-653	8-653#	8-659	8-659#	8-662	8-662#	8-670	8-670#	8-670#	8-672	8-672#	8-678
8-678#	8-683	8-683#	8-703	8-703#	8-707	8-707#	8-717	8-717#	8-722	8-722#	8-725	8-725#	8-725#
8-726	8-726#	8-729	8-729#	8-734	8-734#	8-737	8-737#	8-737#	8-738	8-738#	8-740	8-740#	8-764
8-764#	8-767	8-767#	8-767#	8-770	8-770#	8-784	8-784#	8-790	8-790#	8-793	8-793#	8-793#	8-794
8-794#	8-799	8-799#	8-802	8-802#	8-802#	8-805	8-805#	8-818	8-818#	8-821	8-821#	8-833	8-833#
8-838	8-838#	8-841	8-841#	8-841#	8-842	8-842#	8-846	8-846#	8-865	8-865#	8-868	8-868#	8-872
8-872#	8-877	8-877#	8-880	8-880#	8-887	8-887#	8-887#	8-889	8-889#	8-893	8-893#	8-894	8-894#
8-912	8-912#	8-915	8-915#	8-934	8-934#	8-934#	8-937	8-937#	8-:04	8-:04#	8-:04#	8-:05	8-:05#
8-:11	8-:11#	8-:32	8-:32#	8-:35	8-:35#	8-:55	8-:55#	8-:58	8-:58#	8-:61	8-:61#	8-:65	8-:65#
8-:83	8-:83#	8-:91	8-:91#	8-:91#	8-:92	8-:92#	8-:96	8-:96#	8-:02	8-:02#	8-:04	8-:04#	8-:08
8-:08#	8-:08#	8-:09	8-:09#	8-:11	8-:11#	8-:34	8-:34#	8-:39	8-:39#	8-:42	8-:42#	8-:57	8-:57#
8-:60	8-:60#	8-:63	8-:63#	8-:66	8-:66#	8-:71	8-:71#	8-:74	8-:74#	8-:82	8-:82#	8-:82#	8-:83
8-:83#	8-:90	8-:90#	8-:93	8-:93#	8-:96	8-:96#	8-:99	8-:99#	8-<05	8-<05#	8-<08	8-<08#	8-<17
8-<17#	8-<17#	8-<18	8-<18#	8-<33	8-<33#	8-<34	8-<34#	8-<55	8-<55#	8-<58	8-<58#	8-<61	8-<61#
8-<64	8-<64#	8-<69	8-<69#	8-<72	8-<72#	8-<79	8-<79#	8-<82	8-<82#	8-<82#	8-<82#	8-<82#	8-<82#
8-=04#	8-=07	8-=07#	8-=10	8-=10#	8-=15	8-=15#	8-=18	8-=18#	8-=26	8-=26#	8-=26#	8-=29	8-=29#
8-=48	8-=48#	8-=51	8-=51#	8-=54	8-=54#	8-=57	8-=57#	8-=63	8-=63#	8-=66	8-=66#	8-=73	8-=73#
8-=73#	8-=76	8-=76#	8-=96	8-=96#	8-=99	8-=99#	8->02	8->02#	8->05	8->05#	8->11	8->11#	8->14
8->14#	8->21	8->21#	8->21#	8->24	8->24#	8->46	8->46#	8->49	8->49#	8->87	8->87#	8->91	8->91#

