

RX02

SS PERF EXER
CZRXDA0

AH-E513A-MC
COPYRIGHT © 1978
FICHE 1 OF 1

AUG 1978
digital
MADE IN USA

This microfiche card contains a grid of frames. The first two columns on the left contain header information, including the title 'SS PERF EXER CZRXDA0' and other identifying data. The remaining columns contain individual frames of data, likely performance metrics or test results, arranged in a structured grid format. The data is presented in a way that is readable when viewed through a microfiche reader.

11

REM 8

IDENTIFICATION

PRODUCT CODE: AC-E512A-MC
PRODUCT NAME: CZRADA0 RX02 SS PERF EXER
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: L. S. PRUCHA

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

1.0	GENERAL INFORMATION
1.1	PROGRAM ABSTRACT
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 SIX STEPS OF EXECUTION
2.1.2	SAMPLE RUN-THROUGH
2.2	HOW TO CREATE A CHAINABLE FILE
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
3.0	ERROR INFORMATION
3.1	WRITE ERROR
3.2	CRC ERROR
3.3	NO CRC ERROR BUT DATA ERROR
3.4	CRC ERROR BUT NO DATA ERROR
3.5	SEEK ERROR
3.6	CHECKSUM ERROR
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
6.0	TEST SUMMARIES
6.1	UNIT/DRIVE SELECTION
6.2	DATA PATTERNS
6.3	FUNCTIONAL TESTS
6.4	TRACK SEQUENCING
6.5	SECTOR/TRACK ADDRESSING
6.6	DISKETTE DENSITY
6.7	PROGRAM CONTROL

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THIS PROGRAM EXERCISES TWO RX02 SUBSYSTEMS (FOUR DRIVES), MAINTAINS DRIVE STATISTICS AND PROVIDES RUN SUMMARIES SO THAT SEEK AND DATA ERROR RATES MAY BE DETERMINED. THE PERFORMANCE EXERCISER WILL GIVE THE USER CONFIDENCE, AFTER RUNNING SUCCESSFULLY, THAT THE SYSTEM IS PERFORMING WITHIN SPECIFICATION.

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.)

1.2.2 SOFTWARE REQUIREMENTS

THIS DIAGNOSTIC IS DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR AS DESCRIBED IN PARAGRAPH 2.0.

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP USERS MANUAL MD-11-DZQXA
DIAGNOSTIC SUPERVISOR PROGRAM LISTING

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

NONE

1.5 ASSUMPTIONS

THE HARDWARE OTHER THAN THE SUBSYSTEM BEING TESTED IS ASSUMED TO WORK PROPERLY. FALSE ERRORS MAY BE REPORTED IF THE PROCESSOR, MEMORY, ETC., DO NOT FUNCTION PROPERLY.

2.0 OPERATING INSTRUCTIONS

2.1 HOW TO RUN THIS DIAGNOSTIC

2.1.1 THE SIX STEPS OF EXECUTION

THIS DIAGNOSTIC SHOULD BE LOADED AND STARTED USING NORMAL XXDP PROCEDURES. THE START COMMAND SHOULD NOT SPECIFY AN ADDRESS, BECAUSE THE DIAGNOSTIC HAS THE PROPER TRANSFER ADDRESS CODED INTO IT.

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

* STEP 1 *

A SHORT SERIES OF "HARDCORE QUESTIONS" WILL BE ASKED:

QUESTION -----	MEANING -----
L-CLK (L) N ?	IS THERE AN L-CLOCK?
P-CLK (L) N ?	" " " P-CLOCK?
50HZ (L) N ?	IS THE POWER 50 CYCLES (AS IN EUROPE)?
LSI (L) N ?	IS MACHINE AN LSI?
LPT (L) N ?	IS THERE A LINE PRINTER?
MEM (K) (D) 16 ?	HOW MANY K OF MEMORY ARE THERE?

THE DEFAULTS (SHOWN AFTER EACH QUESTION) CAN BE SELECTED BY HITTING CARRIAGE RETURN. IT IS POSSIBLE THAT NOT ALL OF THE QUESTIONS WILL BE ASKED: FOR EXAMPLE, IF YOU SAY "YES" TO THE L-CLOCK QUESTION, THE P-CLOCK QUESTION WILL NOT BE ASKED.

* STEP 2 *

WHEN YOU HAVE ANSWERED ALL THE HARDCORE QUESTIONS, THE DIAGNOSTIC WILL ISSUE THE PROMPT "DS-B)". 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 "DS-B)" 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:

LOE	LOOP ONE ERROR
HOE	HALT ON ERROR
IER	INHIBIT ERROR PRINTOUT

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

* STEP 3 *

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 4 *

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. DIAGNOSTIC WILL IN THE FUTURE 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.

--> ANSWER THE EXPANSION QUESTION WITH A "CR" OR 0 AND "CR". THIS WORD IS RESERVED FOR FUTURE EXPANSION.

* STEP 5 *

AFTER YOU HAVE ANSWERED ALL THE HARDWARE QUESTIONS 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, 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.

--> AGAIN ANSWER THE EXPANSION QUESTION WITH A "CR".

* STEP 6 *

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 DS-B)).
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 ENLESSLY ON THE BLOCK OF CODE THAT DETECTED THE ERROR.
IF NO ERROR PRINT OUT IS DESIRED, THEN ALSO SET THE IER FLAG (SEE BELOW).

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

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 REISSUED.

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 2, 3, 4, 5, AND 6 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 OCCURED. 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 ON ERROR).

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:

	BY
	WHOM
	ENTERED:
. R DZRKXX	0
DZRKXX	D
L-CLK (L) N ? Y	D,0
SOHZ (L) N ?	D
LSI (L) N ?	D
LPT (L) N ?	D
MEM (K) (D) 16 ?	D
DS-B>STA/PASS: 1/FLAGS: HOE	D,0
# UNITS (D) ? 2	D,0
UNIT 1	D
DRIVE (0) ? 0	D,0
UNIT 2	D
DRIVE (0) ? 1	D,0
CHANGE SW (L) ? N	D,0
DZRKXX HARD ERR 00004 TST 003 SUB 002 PC: 004130	D
ERR HLT	D
DS-B>PRO/FLAGS: IER: LOE: HOE=0	D,0

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	

C	0
DS-B>CON/FLAGS: HOE: IER: LOE=0	D,0

```

CHANGE SW (L) ? N          D,0
DZRKXX EOP 1                D
DS-B>RESTART/PASS: 1       D,0
CHANGE SW (L) ? N          D,0
-----
-----
-----
-----
-----

```

2.2 HOW TO CREATE A CHAINABLE FILE

THE DIAGNOSTIC AS RECEIVED FROM RELEASE ENGINEERING CANNOT BE RUN IN CHAIN MODE. THAT IS WHY IT BEARS THE EXTENSION "BIN" INSTEAD OF "BIC". THERE IS A WAY, HOWEVER, TO CREATE A CHAINABLE PROGRAM FROM WHAT YOU'VE GOT.

IT CONSISTS OF RUNNING THE PROGRAM WITH THE SPECIAL COMMAND "CCI" ISSUED WHERE YOU WOULD NORMALLY ISSUE A START COMMAND (TO THE PROMPT DS-B). THIS COMMAND CAUSES THE DIAGNOSTIC TO GO THRU ALL THE QUESTIONS AND ANSWERS AND THEN TO HALT, JUST WHERE IT WOULD ORDINARILY BEGIN EXECUTION OF THE HARDWARE TEST CODE. AT THIS POINT YOU CAN DUMP THE PROGRAM AS IT SITS IN CORE TO THE LOAD MEDIUM, WITH THE NEW EXTENSION "BIC".

HERE IS A SAMPLE DIALOGUE TO ACCOMPLISH THIS:

```

. R UPD2
RESTART: XXXXXX
*CLR
*LOAD DIAG. BIN
XFER: 200 CORE: 0,60602
*START 200
L-CLK (L) N ?
-----
-----
DS-B>CCI
# UNITS (D) ? 4
-----
-----
CHANGE SW (L) ? N
PTAB END: 60632

```

```

*****
*AT THIS POINT THE MACHINE HALTS AND*
*YOU MUST RESTART AT ADDRESS XXXXXX*
*****

```

```

*HCORE 60632
CORE: 0,60632
*DUMP DKO: DIAG. BIC

```

THE RESULT OF DOING THIS IS THAT YOU CAN NOW BUILD AN XXDP CHAIN FILE

CONTAINING THE XXDP COMMAND

. R DIAG. BIC

AND THE DIAGNOSTIC WILL EXECUTE WITHOUT MANUAL INTERVENTION, USING THE ANSWERS THAT YOU GAVE IT WHEN YOU DID THE CCI COMMAND.

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:

HOW ENTERED	LEGAL COMMANDS
1. OPERATOR ENTERED 'RUN DIAG'	START PRINT DISPLAY FLAGS ZFLAGS
2. DIAGNOSTIC HAS FINISHED ALL ITS REQUESTED PASSED	START RESTART PRINT DISPLAY FLAGS ZFLAGS
3. OPERATOR INTERRUPTED THE DIAGNOSTIC WITH CTRL/C	START RESTART CONTINUE PRINT DISPLAY FLAGS ZFLAGS
4. AN ERROR WAS ENCOUNTERED WITH THE HOE FLAG SET SET	START RESTART CONTINUE PROCEED PRINT DISPLAY FLAGS ZFLAGS

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 EXECUTION. B "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, SUBTEST, 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 TES 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

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 IF GIVEN BECOME THE NEW DEFAULTS. THE COMMAND MAY BE ISSUED WHEN COMMAND 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 REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

THE SWITCH ARGUMENTS ARE AS IN THE START COMMAND EXCEPT:

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

PRO(CEED)/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

CCI/TEST: TEST-LIST/PASS: PASS-CNT/FLAGS: FLAG-LIST/EOP: EOP-INCR

THE DIAGNOSTIC EXECUTES THRU ALL OPERATOR DIALOGUE AND HALTS AT THE HARDWARE TEST CODE. NOW THE OPERATOR CAN DUMP THE CORE IMAGE TO THE MEDIUM WITH A BIC EXTENSION.

THE BIC FILE MUST BE HANDLED DIFFERENTLY DEPENDING ON WHETHER IT IS RUN MANUALLY OR IN CHAIN MODE. IF RUN MANUALLY IT CAN BE INVOKED EITHER WITH A "START" (IN WHICH CASE IT WILL BEHAVE LIKE THE BIN FILE: THE PRE-GENERATED ANSWERS TO OPERATOR QUESTIONS WILL BE IGNORED) OR WITH A "RESTART" (IN WHICH CASE THE PRE-GENERATED OPERATOR ANSWERS WILL BE USED).

IF RUN IN CHAIN MODE, AUTOMATIC EXECUTION WILL COMMENCE IMMEDIATELY FROM THE XXDP COMMAND ".R DIAG". THE COMMAND PROMPT "DS-B)" WILL NOT BE ISSUED.

ANY SWITCHES SPECIFIED ON THE CCI COMMAND WILL CARRY OVER WHEN THE BIC FILE IS RUN IN CHAIN MODE (EXCEPT THAT UAM IS ALWAYS SET THERE) BUT WILL NOT CARRY OVER WHEN IT IS RUN MANUALLY.

TO DO A CCI ON A FULL SIZED DIAGNOSTIC (14.5K WORDS), A MACHINE SIZE LARGER THAN 16K IS REQUIRED. THE EXACT SIZE NEEDED DEPENDS ON WHICH UTILITY IS USED TO EXECUTE THE DIAGNOSTIC AT CCI TIME.

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(MT)

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,
SAY) 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.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE
P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT
VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED
IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE
BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS
EXHAUSTED. THE LAST VALUE THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED
OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT
VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE
PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS
RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

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 TRANLATES 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 64 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 64 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (1,2,3,...,64) EXCEPT FOR UNIT 50, WHICH SHOULD RECEIVE THE VALUE 49. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 20 UNITS AND THE NUMBER 77 FOR THE LAST 44 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

UNITS (D) ? 64

UNIT 1

<QUESTION 1> ? 75
<QUESTION 2> ? 1-20
<QUESTION 3> ? 76

UNIT 21

<QUESTION 1> ?
<QUESTION 2> ? 21-49,,51-64
<QUESTION 3> ? 77

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 64 TABLES. SLOT TWO RECEIVES THE VALUES 1,2,3,...,20 IN TABLES 1 THRU 20 AND A CONSTANT 20 IN TABLES 21 THRU 64. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 64 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 21 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS A CONSTANT 75 IN TABLES 21 THRU 64, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 21,22,23,...,49 IN TABLES 21 THRU 49, AND GETS A 49 IN SLOT 50, AND GETS THE VALUES 51,52,53,...,64 IN TABLES 51 THRU 64. SLOT THREE GETS THE VALUE 77 IN TABLES 21 THRU 64.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 64 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ON QUESTION (NAMELY QUESTION 2).

3.0 ERROR INFORMATION

3.1 WRITE ERROR

A WRITE ERROR IS AN ERROR WHICH OCCURRED DURING EXECUTION OF A WRITE FUNCTION.

READ ERROR

A READ ERROR IS AN ERROR WHICH OCCURRED DURING EXECUTION OF A READ FUNCTION.

3.2 CRC ERROR

THIS ERROR IS DETECTED BY THE DRIVE DURING A READ OPERATION AND ALSO BY THE PROGRAM IF A DATA CHECK IS PERFORMED.

3.3 NO CRC ERROR BUT DATA ERROR - BAD CRC

3.4 CRC ERROR BUT NO DATA ERROR - BAD CRC

THE ABOVE TWO ERRORS ARE DETECTED WHEN THE PROGRAM IS VERIFYING THE DATA READ OFF THE DISKETTE AGAINST THE DATA THAT SHOULD HAVE BEEN READ.

THE DATA PATTERNS WILL BE FORMATTED FOR DOUBLE DENSITY (SINGLE DENSITY) AS SHOWN.

BYTE #
0 <TRACK ADDRESS; BITS 6 - 0>
1 <SECTOR ADDRESS; BITS 4 - 0>

BYTES 2 THROUGH 253 (125) CONTAIN A SELECTED PATTERN.

254(126) <THE SUM OF ALL BYTES 0 - 253(125)>
255(127) <THE NEGATIVE OF 2 TIMES BYTE 254(126)>

3.5 SEEK ERROR

A SEEK ERROR CAN BE DETECTED VIA BYTE #0 IF A CRC, DATA, CHECKSUM ERROR HAS NOT OCCURRED. ALSO THE DRIVE MAY DETECT A SEEK ERROR IF THE DISKETTE HEADER IS NOT RECOGNIZED OR COULD NOT BE FOUND. A PROGRAMMED RECALIBRATE IS ISSUED TO TRY TO CORRECT EACH SEEK ERROR, IF SELECTED DURING PROGRAM DIALOG.

3.6 CHECKSUM ERROR

THE PROGRAM WILL DETECT A CHECKSUM ERROR BY SUMMING ALL THE DATA READ FROM THE DISKETTE AND COMPARING THAT SUM WITH THE CHECKSUM BYTES. A CHECKSUM ERROR RESULTS FROM AN INCORRECT TRANSFER OF DATA INTERNAL TO THE RX21/RX02 SUBSYSTEM.

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS A STATISTICAL REPORT WILL BE PRINTED OUT OF ALL ACCUMULATED ERRORS.

5.0 DEVICE INFORMATION TABLES

***** RX02 REGISTER BITS *****

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RXCS:	ERR	INT	XM	XM	RX2		SID	DEN	TR	IE	DON	DRV	FUN	FUN	FUN	GO
RXWC:	X	X	X	X	X	X	X	X								WORD COUNT
RXBA:	BUS ADDRESS															
RXES:	X	X	X	X	NXM	WC	SID	DRV	DRV	DEL	DSK	DEN	AC	INT	SID	CRC
							OVF	#1	#1	RDY	DAT	DEN	ERR	LOW	DON	RDY
RXTA:	X	X	X	X	X	X	X	X	X	0						TRACK ADDRESS
RXSA:	X	X	X	X	X	X	X	X	X	0	0	0				SECTOR ADDRESS
RXDB:	DATA BUFFER															

6.0 TEST SUMMARIES

6.1 UNIT/DRIVE SELECTION

UNIT AND DRIVE SELECTION WILL BE ACCOMPLISHED BY MODIFYING HARDWARE P-TABLES DURING A START DIALOG.

6.2 DATA PATTERNS

AVAILABLE DATA PATTERNS ARE SELECTED BY MODIFYING THE SOFTWARE P-TABLE DURING START OR RESTART DIALOG. DATA PATTERNS AVAILABLE ARE:

- 0 = DEFAULT TO 7
- 1 = ZEROS
- 2 = ONES
- 3 = FLOATING ZERO
- 4 = FLOATING ONE
- 5 = 125
- 6 = 333
- 7 = RANDOM

6.3 EXERCISE OPTIONS

AVAILABLE EXERCISES ARE SELECTED BY MODIFYING THE SOFTWARE P-TABLE DURING A START OR RESTART DIALOG. EXERCISES AVAILABLE ARE:

- 0 = DEFAULT TO 7
- 1 = WRITE ONLY
- 2 = WRITE/READ
- 3 = WRITE/READ/DATA CHECK
- 4 = READ/DATA CHECK ONLY
- 5 = READ ONLY (CRC CHECK)
- 6 = WRITE/READ/DATA CHECK ON ALTERNATING DRIVES (*)
- 7 = WRITE/READ/DATA CHECK +/-READ/DATA CHECK (**)

(*) TEST 6 WRITES THEN READ CHECKS ANY SELECTED DATA PATTERN USING ANY TRACK SEQUENCE, BUT ONE TRACK AT A TIME. FIRST ON DRIVE 0 THEN DRIVE WHEN BOTH UNIES HAVE ACCESSED THAT TRACK, IT GOES BACK TO UNIT 0 FOR THE NEXT TRACK, ETC.

(**) THE FIRST HALF OF TEST 7 FORCES THE TRACK SEQUENCE TO INCREMENT UP THROUGH ALL TRACKS DOING WRITE/READ/DATA CHECK FUNCTIONS. THIS VERIFIES THAT ALL TRACKS ARE ACCESSABLE. THE SECOND HALF OF THE PASS WILL USE THE SEQUENCE SELECTED BY THE OPERATOR AS INDICATED BELOW, AND ONLY READ AND CHECK THE DATA JUST WRITTEN. THIS VERIFIES THAT THE DATA CAN BE READ FROM A TRACK AFTER THE HEAD HAS BEEN MOVED AWAY FROM AND BACK TO THAT TRACK. AT THE COMPLETION OF THE PASS THE DELETED DATA BIT IN TEST CONDITIONS IS COMPLEMENTED AND THE NEXT PASS WILL BE RUN UNDER THIS NEW CONDITION.

6.4 TRACK SEQUENCING

TRACK SEQUENCE OR TYPE OF HEAD MOVEMENT MAY BE SELECTED BY MODIFYING THE SOFTWARE P-TABLE OF THE DIAGNOSTIC SUPERVISOR. TRACK SEQUENCES AVAILABLE FOR SELECTION ARE:

- 0 = DEFAULT TO 7
- 1 = INCREMENT O. D. UP TO I. D.
- 2 = DECREMENT I. D. DOWN TO O. D.
- 3 = INCREMENT O. D., THEN DECREMENT I. D.
- 4 = BOUNCE BETWEEN O. D. AND I. D.
- 5 = BOUNCE BETWEEN DECREASING I. D. AND INCREASING O. D.
- 6 = BOUNCE BETWEEN O. D. AND DECREASING I. D.
- 7 = RANDOM

O. D. = OUTSIDE DIAMETER (TRACK)
I. D. = INSIDE DIAMETER (TRACK)

6.5 SECTOR/TRACK ADDRESSING

IT WILL BE POSSIBLE TO TEST THE DISKETTES BETWEEN TRACK AND SECTOR ADDRESS LIMITS OTHER THAN BETWEEN THE NORMAL OUTER DIAMETER (OD) AND INNER DIAMETER (ID) TRACK ADDRESSES, AND/OR MINIMUM (FIRST) AND MAXIMUM (LAST) SECTOR ADDRESS, BY MODIFYING THE SOFTWARE P-TABLE DURING A START OR RESTART DIALOG.

6.6 DISKETTE DENSITY

ALL TESTS WILL RUN AT DOUBLE DENSITY UNLESS SELECTED AS SINGLE DENSITY DURING A START OR RESTART DIALOG.

6.7 PROGRAM CONTROL

BEHAVIOR OF THE PERFORMANCE EXERCISOR MAYBE MODIFIED BY USE OF THE FOLLOWING PROGRAM CONTROLS:

- | | |
|---|------------------------|
| 1. HALT ON ERROR | PROVIDED BY SUPERVISOR |
| 2. HALT AT END OF PASS | PROVIDED BY SUPERVISOR |
| 3. DON'T PRINT ERROR MESSAGE | PROVIDED BY SUPERVISOR |
| 4. RETRY ON ERROR. LOG HARD/SOFT ERRORS | SOFTWARE P-TABLE |
| 5. RECALIBRATE ON SEEK ERRORS | SOFTWARE P-TABLE |

7.0 LISTING INDEX

7.1 LISTING

&

```
3003 . TITLE PROGRAM HEADER AND TABLES
3004 . SBTTL PROGRAM HEADER
3038
3040 . ENABL ABS,AMA
3041 002000 = 2000
3043
3044 002000 BGNMOD
3045
3046 ;++
3047 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
3048 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
3049 ;--
3050
3051 002000 POINTER ALL
3052
3060
3061 002000 HEADER CZRXDA0,0,0,2100,2100,2100,,E,TITLX
3062
3068 ;-----
3069 002110 054122 031060 051440 TITLX: .ASCIZ /RX02 SS PERF EXER/
3070 .EVEN
3071 ;-----
```

3074
3075
3076
3077
3078
3079
3080
3081
3082

002132

.SBTTL DISPATCH TABLE

;++

; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.

DISPATCH 1


```
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099 002136          BGNHW  DFPTBL  
3100  
3101 002140 177170  .WORD 177170      ;UNIBUS ADDRESS  
3102 002142 000264  .WORD 264         ;VECTOR ADDRESS  
3103 002144 000000  .WORD 0           ;DRIVE #  
3104 002146 000000  .WORD 0           ;FUTURE EXPANSION  
3105  
3111  
3112 002150          ENDHW
```

```
3115 .SBTTL SOFTWARE P-TABLE
3116
3117 ;++
3118 ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
3119 ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
3120 ;--
3121
3122 002150          BGNSW  SFPTBL
3123
3124 002152 000000  RXXX: .WORD 0          ;FUTURE EXPANSION-RX
3125 002154 000000          .WORD 0          ;CONTROL WORD FOR
3126 002156 000000  TSTN: .WORD 0          ;TEST #
3127 002160 000000  TSTPAT: .WORD 0        ;TEST PATTERN #
3128 002162 000000  TRKSEQ: .WORD 0        ;TRACK SEQUENCE #
3129 002164 000021  SWREG: .WORD 21       ;SOFTWARE SWITCH REG
3130 002166 000000  OTDITK: .WORD 0       ;OUTSIDE DIA. TRACK LIMIT
3131 002170 000114  INDITK: .WORD 114     ;INSIDE DIA. TRACK LIMIT.
3132 002172 000001  MINSEC: .WORD 1       ;MINIMUM SECTOR LIMIT
3133 002174 000032  MAXSEC: .WORD 32      ;MAXIMUM SECTOR LIMIT
3134
3141
3142 002176          ENDSW
3143
3144 002176          ENDMOD
```

3157
3158
3195
3205
3206 002176
3207
3208
3209
3210
3211
3212
3213 002176

.TITLE GLOBAL AREAS
.SBTTL GLOBAL EQUATES SECTION

BGNMOD

;++
; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
; ARE USED IN MORE THAN ONE TEST.
;--

EQUALS

; BIT DIFINITIONS

(1)		BIT15== 100000
(1)		BIT14== 40000
(1)		BIT13== 20000
(1)		BIT12== 10000
(1)		BIT11== 4000
(1)		BIT10== 2000
(1)		BIT09== 1000
(1)		BIT08== 400
(1)		BIT07== 200
(1)		BIT06== 100
(1)		BIT05== 40
(1)		BIT04== 20
(1)		BIT03== 10
(1)		BIT02== 4
(1)		BIT01== 2
(1)		BIT00== 1

(1)	100000	
(1)	040000	
(1)	020000	
(1)	010000	
(1)	004000	
(1)	002000	
(1)	001000	
(1)	000400	
(1)	000200	
(1)	000100	
(1)	000040	
(1)	000020	
(1)	000010	
(1)	000004	
(1)	000002	
(1)	000001	

BIT9== BIT09
BIT8== BIT08
BIT7== BIT07
BIT6== BIT06
BIT5== BIT05
BIT4== BIT04
BIT3== BIT03
BIT2== BIT02
BIT1== BIT01
BIT0== BIT00

; EVENT FLAG DEFINITIONS
; EF32: EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
; EF16: EF01 AVAILABLE FOR PROGRAM USE

(1)	000040	EF. START== 32.
(1)	000037	EF. RESTART== 31.
(1)	000036	EF. CONTINUE== 30.
(1)	000035	EF. NEW== 29.
(1)	000034	EF. PWR== 28.
(1)		
(1)	000020	EF16== 16.
(1)	000017	EF15== 15.
(1)	000016	EF14== 14.

; START COMMAND WAS ISSUED
; RESTART COMMAND WAS ISSUED
; CONTINUE COMMAND WAS ISSUED
; A NEW PASS HAS BEEN STARTED
; A POWER-FAIL/POWER-UP OCCURRED

(1)	000015	EF13==	13.
(1)	000014	EF12==	12.
(1)	000013	EF11==	11.
(1)	000012	EF10==	10.
(1)	000011	EF09==	9.
(1)	000010	EF08==	8.
(1)	000007	EF07==	7.
(1)	000006	EF06==	6.
(1)	000005	EF05==	5.
(1)	000004	EF04==	4.
(1)	000003	EF03==	3.
(1)	000002	EF02==	2.
(1)	000001	EF01==	1.
(1)		;	
(1)		;	PRIORITY LEVEL DEFINITIONS
(1)		;	
(1)	000340	PR107==	340
(1)	000300	PR106==	300
(1)	000240	PR105==	240
(1)	000200	PR104==	200
(1)	000140	PR103==	140
(1)	000100	PR102==	100
(1)	000040	PR101==	40
(1)	000000	PR100==	0
3214			
3215		;	
3216		;	BIT DEFINITIONS
3217		;	
3218	100000	BIT15==	100000
3219	040000	BIT14==	40000
3220	020000	BIT13==	20000
3221	010000	BIT12==	10000
3222	004000	BIT11==	4000
3223	002000	BIT10==	2000
3224	001000	BIT09==	1000
3225	000400	BIT08==	400
3226	000200	BIT07==	200
3227	000100	BIT06==	100
3228	000040	BIT15==	40
3229	000020	BIT04==	20
3230	000010	BIT03==	10
3231	000004	BIT02==	4
3232	000002	BIT01==	2
3233	000001	BIT00==	1
3234		;	
3235	001000	BIT9==	BIT09
3236	000400	BIT8==	BIT08
3237	000200	BIT7==	BIT07
3238	000100	BIT6==	BIT06
3239	000040	BIT5==	BIT05
3240	000020	BIT4==	BIT04
3241	000010	BIT3==	BIT03
3242	000004	BIT2==	BIT02
3243	000002	BIT1==	BIT01
3244	000001	BIT0==	BIT00
3245		;	

```
3246 ;EVENT FLAG DEFINITIONS
3247 ; EF32: EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
3248 ; EF16: EFO1 AVAILABLE FOR PROGRAM USE
3249 000040 EF. START== 32. ;START COMMAND WAS ISSUED.
3250 000037 EF. RESTART== 31. ;RESTART COMMAND WAS ISSUED.
3251 000036 EF. CONTINUE== 30. ;CONTINUE COMMAND WAS ISSUED.
3252 000035 EF. NEW== 29. ;A NEW PASS HAS BEEN STARTED.
3253 000034 EF. PWR== 28. ;A POWER FAIL/POWER-UP OCCURRED
3254 ;
3255 000020 EF16== 16.
3256 000017 EF15== 15.
3257 000016 EF14== 14.
3258 000015 EF13== 13.
3259 000014 EF12== 12.
3260 000013 EF11== 11.
3261 000012 EF10== 10.
3262 000011 EF09== 9.
3263 000010 EF08== 8.
3264 000007 EF07== 7.
3265 000006 EF06== 6.
3266 000005 EF05== 5.
3267 000004 EF04== 4.
3268 000003 EF03== 3.
3269 000002 EF02== 2.
3270 000001 EF01== 1
3271 ;
3272 ;PRIORITY LEVEL DEFINITIONS
3273 ;
3274 000340 PR107== 340
3275 000300 PR106== 300
3276 000240 PR105== 240
3277 000200 PR104== 200
3278 000140 PR103== 140
3279 000100 PR102== 100
3280 000040 PR101== 40
3281 000000 PR100== 0
3282 ;
3283 ;PROGRAM DEFINITIONS
3284 ;
3285 000200 TRBIT==200
3286 000040 DNBIT==40
```

```

3300 .SBTTL GLOBAL DATA SECTION
3301
3302 ;++
3303 ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
3304 ; IN MORE THAN ONE TEST.
3305 ;--
3306
3307 ;
3308 ; STORAGE FOR DEVICE REGISTERS
3309 ;
3310 002176          DEVREG 0,0,DEVDAT,REGMSK
3311
3312 ;-----
3313 002204 000000  RESTAR: .WORD 0          ;RESTART FLAG
3314 002206 000000  ABORT: .WORD 0          ;ABORT FLAG
3315 002210 000000  SDD: .WORD 0          ;SYSTEM DRIVES DONE (SEE REG. DEF. BELOW)
3316 002212 000000  SUT: .WORD 0          ;SYSTEM UNDER TEST (SEE REG. DEF. BELOW)
3317 002214 000000  UUT: .WORD 0          ;UNIT UNDER TEST (SEE REG. DEF. BELOW)
3318 002216 000000  UUTADR: .WORD 0        ;UUT UNIBUS ADR
3319 002220 000000  UUTOFF: .WORD 0       ;UUT ADDRESSING OFFSET
3320 002222 000000  DEN: .WORD 0          ;DENSITY FLAG
3321 002224 000000  DELDAT: .WORD 0       ;DELETED DATA FLAG
3322 002226 000000  UOADR: .WORD 0        ;UNIT 0 ADR
3323 002230 000000  U1ADR: .WORD 0        ;UNIT 1 ADR
3324 002232 000000  SYSERR: .WORD 0       ;SYSTEM ERROR
3325 002234 000000  ERRTP: .WORD 0        ;ERROR TYPE
3326 002236 000000  CSRUUT: .WORD 0       ;CONT/STATUS REG UUT
3327 002240 000000  ESRUUT: .WORD 0       ;ERROR/STATUS REG UUT
3328 002242 000000  RETRY: .WORD 0        ;//(10)DATART/(4)RDRT/(2)WTRT/(1)SEEK/ SEE BELOW
3329 002244 000000  SEEKRT: .WORD 0       ;SEEK RETRY COUNT
3330 002246 000000  CKSMRT: .WORD 0       ;CHECK SUM RETRY COUNT
3331 002250 000000  CRCBRT: .WORD 0       ;CRC BAD RETRY COUNT
3332 002252 000000  CRCERT: .WORD 0       ;CRC ERR RETRY COUNT
3333 002254 000000  DATART: .WORD 0       ;DATA RETRY COUNT
3334 002256 000000  DARDRT: .WORD 0       ;DATA READ RETRY COUNT
3335 002260 000000  DAWTRT: .WORD 0       ;DATA WRITE RETRY COUNT
3336 002262 000000  READRT: .WORD 0       ;READ RETRY COUNT
3337 002264 000000  WRTRT: .WORD 0        ;WRITE RETRY COUNT
3338 002266 000000  DDERCT: .WORD 0       ;D. D. ERR RETRY COUNT
3339 002270 000000  WDCNT: .WORD 0        ;WORD COUNT
3340 002272 000000  TRACK: .WORD 0        ;TRACK ADR
3341 002274 000000  SECTOR: .WORD 0       ;SECTOR ADR
3342 002276 000000  TRKDN: .WORD 0        ;TRACK DONE (UUT) FLAG
3343 002300 000000  SECDN: .WORD 0        ;SECTOR DONE (UUT) FLAG
3344 002302 000000  UOLECT: .WORD 0       ;UNIT 0 VECTOR
3345 002304 000000  U1LECT: .WORD 0       ;UNIT 1 VECTOR
3346 002306 000000  HARDER: .WORD 0       ;HARD ERROR
3347 002310 000000  PRTECD: .WORD 0       ;PRINT ERR CODE FLAG

```

3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388

```

-----
***** SOFTWARE REGISTER DEFINITIONS *****
-----
          BIT#
          03! 02! 01! 00
-----+-----+-----+-----+
SDD:    ! 11! 10! 01! 00!  <- UUT CODES-EQUIV TO A BIT SET IN THIS REG
&      -+--+--+--+--+    -THAT IS UUT=00 IS SDD BIT#0 SET
SUT:    ! 11! 10! 01! 00!  <- UUT CODES-
-----+-----+-----+-----+
-----
          ! RX02          ! RXXX-FUTURE EXPANSION
-----+-----+-----+-----+
UUT:    00 = UNIT#0/DRV#0    SIDE#0/DRV#0
        01 = UNIT#0/DRV#1    SIDE#0/DRV#1
        10 = UNIT#1/DRV#0    SIDE#1/DRV#0
        11 = UNIT#1/DRV#1    SIDE#1/DRV#1
-----+-----+-----+-----+
          !!
          !!---<DRIVE #
          !!---<UNIT # (RX02) OR SIDE # (RXXX)
-----+-----+-----+-----+
          15! 14! 13! 12! 11! 10! 09! 08! 07! 06! 05! 04! 03! 02! 01! 00!
-----+-----+-----+-----+-----+-----+-----+-----+-----+
ERRRTP: !ERR!ERR!DON!ITR!WRT!RD!FIL!UNK!  !DD!DD!  !  !CK!
          !BIT!NOT!NO!NO!ERR!ERR!EMP!ERR! - !MIS!UNX! - !DAT!SUM!CRC!SEK!
          !SET!ITR!DON!  !ERR!  !  !  !  !  !  !  !
-----+-----+-----+-----+-----+-----+-----+-----+-----+
          !  !  !DRV!  !  !WRONG!DON!SID!DRV!NO DONE! FUNCTION
SYSERR: !UNR! TO!DEN!DEN!SYS!DAG!-----!FUN!RDY!RDY!-----! CAUSING
          !ERR!ERR!ERR!ERR!ERR!ERR!SID!DRV! #2!ERR!ERR!FUN!INT! ERROR
-----+-----+-----+-----+-----+-----+-----+-----+-----+
          !  !  !  !CRC!DAT!RD!WRT!SEK!
RETRY:  !  !  !RT!RT!RT!RT!RT!
-----+-----+-----+-----+-----+-----+-----+-----+

```

NOTE: RXXX IS REFERENCE FOR FURTHER EXPANSION

3390
3391
3392 002312 000
3393 002313 000
3394 002314 000
3395 002315 000
3396 002316 000
3397 002317 000
3398 002320 000
3399 002321 000
3400
3407
3408

```
. SBTTL          READ ERROR CODE BUFFER
;-----;
XERUUT: . BYTE  0          ;ERROR CODE UUT
WC:      . BYTE  0          ;WORD COUNT UUT
CTKO:    . BYTE  0          ;CUR TRK DRV#0
CTK1:    . BYTE  0          ;CUR TRK DRV#1
TTRK:    . BYTE  0          ;TARGET TRK
TSEC:    . BYTE  0          ;TARGET SEC
SFTSTS:  . BYTE  0          ;MICRO CODE SOFT STATUS
BTRK:    . BYTE  0          ;BAD TRK ADR
;-----;
```


3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3435
3436
3437
3438
3439
3446
3447

002322

.SBTTL GLOBAL TEXT SECTION
; ++
; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
; MORE THAN ONE TEST.
; --
;
; NAMES OF DEVICES SUPPORTED BY PROGRAM
; DEV TYP RX02+
;
; FORMAT STATEMENTS USED IN PRINT CALLS
;

3457
3458
3459
3460
3461
3462
3463
3464
3465
3466 002330
3467
3473
3474
3475
3476
3477
3483
3490
3491
3492
3493 002330

.SBTTL GLOBAL ERROR REPORT SECTION

;++
; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
; THAT ARE USED IN MORE THAN ONE TEST. IT ALSO INCLUDES THE ASCII MESSAGES
; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
;--

BGNMSG

; BIT-NAMES FOR THE DEVICE REGISTERS
;

.EVEN

ENDMSG

3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3512
3518
3525
3531
3538
3547
3555
3561
3562
3569
3575
3576
3577 002332 012700 000001
3578 002336 063700 002420
3579 002342 063700 002422
3580 002346 042700 170000
3581 002352 000241
3582 002354 006100
3583 002356 006100
3584 002360 010037 002420
3585 002364 005000
3586 002366 013700 002422
3587 002372 006000
3588 002374 006000
3589 002376 063700 002420
3590 002402 042700 170000
3591 002406 010037 002422
3592 002412 010037 002424
3593 002416 000207
3594
3595 002420 000000
3596 002422 000000
3597 002424 000000
3598

.SBTTL GLOBAL SUBROUTINES SECTION

;++
; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
; THAT ARE USED IN MORE THAN ONE TEST.
;--

;++
; FUNCTIONAL DESCRIPTION:
; SUBROUTINE TO...
; INPUTS: NONE
; IMPLICIT INPUTS: NONE
; OUTPUTS: RANUM
; IMPLICIT OUTPUTS: NONE
; SUBORDINATE ROUTINES USED: NONE
; FUNCTIONAL SIDE EFFECTS: NONE
; CALLING SEQUENCE: SUB
;--

.SBTTL MOD U. 1.0 - RANDOM GENERATOR

;----- RANDOM GENERATOR -----
RANGEN: MOV #1,RO
ADD RAN1,RO
ADD RAN2,RO
BIC #170000,RO
CLC
ROL RO
ROL RO
MOV RO,RAN1
CLR RO
MOV RAN2,RO
ROR RO
ROR RO
ADD RAN1,RO
BIC #170000,RO
MOV RO,RAN2
MOV RO,RANUM
RTS PC

RAN1: 0
RAN2: 0
RANUM: 0

```

3600          .SBTTL          MOD U. A. 1 - CONVERSION UUT CODE --> SUTPTR
3601          ;-----;
3602
3603 002426 000240          CVUTST: NOP          ;
3604 002430 005037 002530          CLR          SUTCV          ; CLEAR SYS UNDER TEST CONVERTED
3605 002434 032737 000001 002526          BIT          #1,CVUNIT          ; IF DRIVE #0.
3606 002442 001014          BNE          25          ; SELECTED, THEN
3607 002444 032737 000002 002526          BIT          #2,CVUNIT          ; IF UNIT #0 OR RX04 SIDE #0.
3608 002452 001004          BNE          15          ; THEN
3609 002454 052737 000001 002530          BIS          #1,SUTCV          ; SET FOR UNIT CODE=00 IN SUT WORD
3610 002462 000420          BR          ENDCVT          ; BR TO END
3611 002464 052737 000004 002530 15:          BIS          #4,SUTCV          ; ELSE, SET FOR UNIT CODE=10 IN SUT WORD
3612 002472 000414          BR          ENDCVT          ; BR TO END
3613 002474 032737 000002 002526 25:          BIT          #2,CVUNIT          ; IF UNIT #0 OR RX04 SIDE #0.
3614 002502 001004          BNE          35          ; THEN
3615 002504 052737 000002 002530          BIS          #2,SUTCV          ; SET FOR UNIT CODE=01 IN SUT WORD
3616 002512 000404          BR          ENDCVT          ; BR TO END
3617 002514 052737 000010 002530 35:          BIS          #10,SUTCV          ; ELSE, SET FOR UNIT CODE=11 IN SUT WORD
3618 002522 000240          NOP          ;
3619 002524 000207          ENDCVT: RTS          PC          ; RETURN
3620          ;-----;
3621 002526 000000          CVUNIT: 0          ; UNIT CODE TO BE CONVERTED
3622 002530 000000          SUTCV: 0          ; SYS UNDER TEST AS CONVERTED
3623          ;MOD U. A. 1 ----- END MODULE -----;
3624
3625          .SBTTL          MOD U. A. 2 - CONVERSION SUTPTR --> UUT CODE
3626          ;-----;
3627
3628 002532 013705 015644          CVSTUT: MOV          SUTPTR,R5          ; SAVE SUT POINTER IN R5
3629 002536 005004          CLR          R4          ; CLEAR R4 (RESET UNIT CODE)
3630 002540 032705 000001          15:          BIT          #1,R5          ; IF LSB R5
3631 002544 001003          BNE          25          ; EQUALS 1, THEN BR TO 25
3632 002546 006205          ASR          R5          ; SHIFT RIGHT R5
3633 002550 005204          INC          R4          ; INCREMENT R4
3634 002552 000772          BR          15          ; BR TO 15
3635 002554 010437 002600          25:          MOV          R4,UNITST          ; THEN R4 CONTAINS UUT CODE
3636 002560 006304          ASL          R4          ; DOUBLE UNIT CODE FOR ADR
3637 002562 010437 002220          MOV          R4,UUTOFF          ; SET UUT OFFSET
3638 002566 062704 007356          ADD          #UTOO,R4          ; GET UUT UNIT# FOR PRINT
3639 002572 011437 006620          MOV          (R4),UNIT          ; SET UNIT=PRINT UNIT#
3640 002576 000207          RTS          PC          ; RETURN
3641          ;-----;
3642 002600 000000          UNITST: 0          ;
3643          ;MOD 2. 0A ----- END MODULE -----;
3644
3645 002602          ENDMOD
3646

```

3659
 3660
 3697
 3698 002602
 3699
 3700
 3701
 3702
 3703
 3704
 3705
 3706 002602
 3707 002602 000240
 3708 002604 012737 003354 003140
 3709 002612 012737 003436 003142
 3710 002620 004737 003046
 3711 002624 000240
 3712 002626 005037 003272
 3713 002632 005037 003300
 3714 002636 012702 004442
 3715 002642 012701 003506
 3716 002646 012737 000023 003274
 3717 002654 004737 003144
 3718 002660 000240
 3719 002662 012737 003363 003140
 3720 002670 012737 003436 003142
 3721 002676 004737 003046
 3722 002702 000240
 3723 002704 012737 000001 003272
 3724 002712 012737 000001 003300
 3725 002720 012702 004672
 3726 002724 012701 003455
 3727 002730 012737 000027 003274
 3728 002736 012737 003455 003276
 3729 002744 000240
 3730 002746 004737 003144
 3731 002752 000240
 3732 002754 000240
 3733 002756 012737 003410 003140
 3734 002764 012737 003436 003142
 3735 002772 004737 003046
 3736 002776 000240
 3737 003000 005037 003272
 3738 003004 012737 000001 003300
 3739 003012 012702 005156
 3740 003016 012737 000115 003274
 3741 003024 012737 003471 003276
 3742 003032 004737 003144
 3743 003036 000240
 3744 003040
 3745
 3746 003042 000000
 3747 003044 000000
 3748

. TITLE MISCELLANEOUS SECTIONS
 . SBTTL REPORT CODING SECTION

BGNMOD

;++
 ; THE REPORT CODING SECTION CONTAINS THE
 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS
 ;--

```

-----
REPORT: BGNRPT
        NOP
        MOV      #PT19SP,PRT1 ; SETUP REPORT HEADER PART 1
        MOV      #PTUNT1,PRT2 ;
        CALL     PRTHDR       ; PRINT HEADER
        NOP
        ; SETUP DATA PART 1
        CLR      LINECT       ; ZERO LINE COUNTER
        CLR      PRNUM        ; CLEAR PRINT MODE
        MOV      #CKSML,R2    ; SET BEGIN ADR OF DATA-PART 1
        MOV      #PRIDXX,R1   ; SET BEGIN ADR OF TABLE LABELS-PART 1
        MOV      #19.,LINES   ; SET # OF LINES TO PRINT
        CALL     PRDAT        ; PRINT DATA
        NOP
        ; SETUP HEADER PART 2
        MOV      #PTEC,PRT1   ;
        MOV      #PTUNT1,PRT2 ;
        CALL     PRTHDR       ; PRINT HEADER
        NOP
        ; SETUP DATA PART 2
        MOV      #1.,LINECT   ; SET LINE COUNTER=1
        MOV      #1.,PRNUM    ; SET PRINT MODE=1
        MOV      #ECLOG,R2    ; SET BEGIN ADR ERROR CODE DATA-PART 2
        MOV      #PTECN,R1    ; SET ERROR CODE PRINT-FORMATED MSG-PART 2
        MOV      #23.,LINES   ; SET # OF LINES TO PRINT
        MOV      #PTECN,LINTYP ;
        CALL     PRDAT        ; PRINT DATA
        NOP
        ;
        MOV      #PTTK,PRT1   ; SETUP HEADER PART 3
        MOV      #PTUNT1,PRT2 ;
        CALL     PRTHDR       ; PRINT HEADER
        NOP
        ;
        CLR      LINECT       ;
        MOV      #1.,PRNUM    ;
        MOV      #TKXX,R2     ; SETUP DATA PART 3
        MOV      #77.,LINES   ;
        MOV      #PTTKN,LINTYP ;
        CALL     PRDAT        ; PRINT DATA PART 3
        ENDRPT: NOP
        ENDRPT
  
```

```

-----
UTTST: 0 ; UNIT #
UTCNT: 0 ; UNIT COUNT
-----
  
```

```

3750          .SBTTL          PRINT REPORT HEADER
3751          ;-----;
3752 003046 005003          PRTHDR: CLR      R3          ;
3753 003050 000240          NOP          ;
3754 003052 013705 003140          MOV      PRT1,R5          ; SETUP 1ST PART OF HEADER PRINT
3755 003056 004737 003332          CALL     PREPT2          ; PRINT 1ST PART
3756 003062 012737 007356 003042          MOV      #UTOO,UTTST          ; GET BEGIN ADR OF UNITS-->TESTED FLAGS
3757 003070 012737 000003 003044          MOV      #3,UTCNT          ; SET UNIT COUNTER
3758 003076 005777 177740          15:    TST      @UTTST          ; IF UNIT TESTED FLAG
3759 003102 100407          BMI      25          ; NOT=-1, THEN
3760 003104 017737 177732 003330          MOV      @UTTST,PAR          ; SET UNIT TESTED # FOR PRINT
3761 003112 013705 003142          MOV      PRT2,R5          ; SET UNIT MSG
3762 003116 004737 003302          CALL     PREPT1          ; PRINT UNIT #
3763 003122 062737 000002 003042          25:    ADD      #2,UTTST          ; ADVANCE ADR OF UNIT TESTED FLAG
3764 003130 005337 003044          DEC      UTCNT          ; DECREMENT UNIT COUNT
3765 003134 001360          BNE      15          ; IF UNIT COUNT=0, THEN
3766 003136 000207          RTS      PC          ; RETURN
3767          ;-----;
3768 003140 000000          PRT1:   0          ;
3769 003142 000000          PRT2:   0          ;
3770          ;-----;
3771          .SBTTL          PRINT REPORT DATA
3772          ;-----;
3773          ;
3774 003144 000240          PRDAT:  NOP          ;
3775 003146 005737 003300          15:    TST      PRNUM          ; IF MODE
3776 003152 001410          BEQ      25          ;
3777 003154 013737 003272 003330          MOV      LINECT,PAR          ; SETUP LINE # TO PRINT
3778 003162 013705 003276          MOV      LINTYP,R5          ; SETUP LINE TYPE TO PRINT
3779 003166 004737 003302          CALL     PREPT1          ; PRINT LINE #
3780 003172 000403          BR       35          ;
3781 003174 012105          25:    MOV      (R1)+,R5          ; SETUP LOG TITLE ADR
3782 003176 004737 003332          CALL     PREPT2          ; PRINT LOG TITLES
3783 003202 012737 007356 003042          35:    MOV      #UTOO,UTTST          ; GET UNIT # FOR PRINT
3784 003210 012737 000004 003044          MOV      #4,UTCNT          ; SETUP UNIT COUNT
3785 003216 012237 003330          45:    MOV      (R2)+,PAR          ; SETUP DATA TO PRINT
3786 003222 005777 177614          TST      @UTTST          ; IF UNIT # NOT = -1
3787 003226 100404          BMI      55          ; THEN
3788 003230 012705 003426          MOV      #PTDAT1,R5          ; SETUP TO PRINT
3789 003234 004737 003302          CALL     PREPT1          ; PRINT DATA
3790 003240 062737 000002 003042          55:    ADD      #2,UTTST          ; SETUP TO CK NEXT UNIT
3791 003246 005337 003044          DEC      UTCNT          ; DECREMENT UNIT COUNT
3792 003252 001361          BNE      45          ; IF DONE ALL UNITS THEN
3793 003254 005237 003272          INC      LINECT          ; INCREMENT LINE COUNT
3794 003260 023737 003274 003272          CMP      LINES,LINECT          ; IF DONE ALL
3795 003266 101327          BHI      15          ; LINES, THEN
3796 003270 000207          RTS      PC          ; RETURN
3797          ;-----;
3798 003272 000000          LINECT: 0          ; LINE COUNTER
3799 003274 000000          LINES:  0          ; # OF LINES TO PRINT
3800 003276 000000          LINTYP: 0          ; LINE PRINT TYPE.
3801 003300 000000          PRNUM:  0          ; PRINT MODE
3802          ;-----;
    
```

```

3804
3805
3806 003302 000240
3807 003304
3808 003326 000207
3809
3810 003330 000000
3811
3812
3813
3814
3815
3816
3817
3818
3819 003332 000240
3820 003334
3821 003352 000207
3822
3823 003354 047045 051445 034461
3824 003363 045 022516 022516
3825 003410 047045 047045 040445
3826 003426 040445 020040 042045
3827 003436 040445 052440 044516
3828 003455 045 022516 031117
3829 003471 045 022516 030523
3830 003506
3831
    
```

```

; SBTTL PRINT REPORT TYPE 1
;-----
PREPT1: NOP ;
PRINTS R5, PAR ;
RTS PC ;
;-----
PAR: 0 ;
;-----
    
```

```

; SBTTL PRINT REPORT TYPE 2
;-----
PREPT2: NOP
PRINTS R5
RTS PC
;-----
PT19SP: .ASCIZ /%N%S19/
PTEC: .ASCIZ /%N%N%RERR%N%ACODE# /
PTTK: .ASCIZ /%N%N%ATRACK# /
PTDAT1: .ASCIZ /%A %D6/
PTUNT1: .ASCIZ /%A UNIT#%D1%A /
PTECN: .ASCIZ /%N%02%AD%S3/
PTTKN: .ASCIZ /%N%S1%D2%S3/
.EVEN
;-----
    
```

3833
 3834 003506 003554
 3835 003510 003603
 3836 003512 003632
 3837 003514 003661
 3838 003516 003710
 3839 003520 003737
 3840 003522 003766
 3841 003524 004015
 3842 003526 004044
 3843 003530 004073
 3844 003532 004122
 3845 003534 004151
 3846 003536 004200
 3847 003540 004227
 3848 003542 004256
 3849 003544 004305
 3850 003546 004334
 3851 003550 004363
 3852 003552 004412

PRIDXX: . WORD PRID01
 . WORD PRID02
 . WORD PRID03
 . WORD PRID04
 . WORD PRID05
 . WORD PRID06
 . WORD PRID07
 . WORD PRID08
 . WORD PRID09
 . WORD PRID10
 . WORD PRID11
 . WORD PRID12
 . WORD PRID13
 . WORD PRID14
 . WORD PRID15
 . WORD PRID16
 . WORD PRID17
 . WORD PRID18
 . WORD PRID19

3853
 3854
 3855
 3856 003554 047045 040445 044103
 3857 003603 045 022516 043101
 3858 003632 047045 040445 047516
 3859 003661 045 022516 044501
 3860 003710 047045 040445 047111
 3861 003737 045 022516 051501
 3862 003766 047045 040445 051103
 3863 004015 045 022516 041501
 3864 004044 047045 040445 042522
 3865 004073 045 022516 053501
 3866 004122 047045 040445 040504
 3867 004151 045 022516 042101
 3868 004200 047045 040445 051110
 3869 004227 045 022516 044101
 3870 004256 047045 040445 051110
 3871 004305 045 022516 044101
 3872 004334 047045 040445 051110
 3873 004363 045 022516 044101
 3874 004412 047045 040445 051110
 3875 004442
 3876

PRID01: . ASCII /XNXACHECK SUM: /
 PRID02: . ASCII /XNXAFILL-EMP BUFF LOG: /
 PRID03: . ASCII /XNXANO ERR BIT: /
 PRID04: . ASCII /XNXAINTER-NO DONE ERR: /
 PRID05: . ASCII /XNXAINTERRUPT ERR: /
 PRID06: . ASCII /XNXASEEK: /
 PRID07: . ASCII /XNXACRC ERR: /
 PRID08: . ASCII /XNXACRC BAD: /
 PRID09: . ASCII /XNXAREAD ERR: /
 PRID10: . ASCII /XNXAWRITE ERR: /
 PRID11: . ASCII /XNXADATA ERR: /
 PRID12: . ASCII /XNXADEL DATA ERR: /
 PRID13: . ASCII /XNXAHRD SEEK: /
 PRID14: . ASCII /XNXAHRD CRC ERR: /
 PRID15: . ASCII /XNXAHRD CRC BAD: /
 PRID16: . ASCII /XNXAHRD READ: /
 PRID17: . ASCII /XNXAHRD WRITE: /
 PRID18: . ASCII /XNXAHRD DATA: /
 PRID19: . ASCII /XNXAHRD DEL. DATA ERR: /
 . EVEN

3878
 3879
 3880
 3881
 3882
 3883
 3884
 3885
 3886
 3887
 3888
 3889
 3890
 3891
 3892
 3893
 3894
 3895
 3896
 3897
 3898
 3899
 3900
 3901
 3902
 3903
 3904
 3916
 3917
 3918

.SBTTL STATISTICAL TABLES

```

;-----;
CKSML: .BLKW 4 ;CKSUM LOG
BUFERL: .BLKW 4 ;FILL/EMPTY BUFFER ERROR LOG
NOERL: .BLKW 4 ;NO ERR BIT LOG
UKINT: .BLKW 4 ;INTERRUPT - NO DONE LOG
INTER: .BLKW 4 ;INTERRUPT ERR
SEK: .BLKW 4 ;SEEK ERR
CRC: .BLKW 4 ;CRC ERR
CRCBAD: .BLKW 4 ;CRC BAD ERR
RD: .BLKW 4 ;READ ERR
WRT: .BLKW 4 ;WRITE ERR
DATA: .BLKW 4 ;DATA ERR
DLTER: .BLKW 4 ;DEL DATA ERR
HSEK: .BLKW 4 ;HARD SEEK ERR
HCRC: .BLKW 4 ;HARD CRC ERR
HCRCBD: .BLKW 4 ;HARD CRC BAD ERR
HRD: .BLKW 4 ;HARD READ ERR
HWRT: .BLKW 4 ;HARD WRITE ERR
HDATA: .BLKW 4 ;HARD DATA ERR
HDD: .BLKW 4 ;HARD DEL DATA ERR
ECLOG: .BLKW 90 ;ERROR CODE LOG
TKXX: .BLKW 308 ;TRACK ERR LOG
;-----;
    
```

ENDST: .WORD 0 ;END TABLE

.EVEN

3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3968
3969
3970
3982
3983
3984
3985
3986
3987
3988
3989

006330
006330
005037 002204
005037 002212
023727 002012 000004
003046
012737 000001 002204
012737 177777 006620
012737 177777 007356
012737 177777 007360
012737 177777 007362
012737 177777 007364
062737 000001 006620
023737 002012 006620
001425
004737 006674
000760
012737 000001 002206
005737 002304
001001
000413
000240
006612
006616 000000
006620 177777
006622 047045 040445 047117
006672

```

.SBTTL INITIALIZE SECTION
; ++
; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE BEGINNING OF EACH PASS.
; --

      BGNINIT
INIT:  REDEF  #EF, CONTINUE
      BCOMPLETE FIN
      CLR   RESTART          ; CLEAR RESTART FLAG
      CLR   SUT              ; CLEAR SYS UNDER TST WORD
      CMP   LSUNIT, #4
      BGT   INITER
      REDEF  #EF, RESTART
      BNCOMPLETE SETUP
      MOV   #1, RESTART      ; SET RESTART FLAG
SETUP: MOV   #-1, UNIT
      MOV   #-1, UTO0        ; RESET UNIT#1
      MOV   #-1, UTO1        ; RESET UNIT#2
      MOV   #-1, UTIO        ; RESET UNIT#3
      MOV   #-1, UT11        ; RESET UNIT#4
15:    ADD   #1, UNIT
      CMP   LSUNIT, UNIT
      BEQ   FIN
      GPHARD UNIT, PLOC
      BNCOMPLETE 15
      JSR   PC, UNPKHP
      BR    15
INITER: PRINTF #INTER1
      MOV   #1, ABORT
FIN:   SETVEC UOUCT, #INTHO, #PRIO7
      TST   U1VECT
      BNE   15
      BR    25
15:    SETVEC U1VECT, #INTH1, #PRIO7
25:    NOP

      EXIT  INIT

-----
PLOC:  .WORD  0
UNIT:  .WORD -1
-----
INTER1: .ASCIZ  /%N%ONLY FOUR UNITS ALLOWED, START OVER/
        .EVEN
-----
      ENDINIT

```

```

3991                                     .SBTTL MOD 1.1 - UNPACK HARDWARE P-TABLES
3992                                     ;-----
3993
3994 006674 000240 UNPKHP: NOP ;
3995 006676 005037 007354 CLR UNT ;CLEAR UNT
3996 006702 013701 006616 MOV PLOC,R1 ;SAVE P-TABLE LOCATION
3997 006706 005737 006620 IFA11: TST UNIT ;IF UNIT
3998 006712 001005 BNE IFB11 ;IS ZERO
3999 006714 012137 002226 MOV (R1)+,UOADR ;LOAD UNIT #0 ADR
4000 006720 012137 002302 MOV (R1)+,UOJECT ;LOAD UNIT #0 VECTOR
4001 006724 000426 BR EIA11 ;BR TO END IF 'A'
4002 006726 021137 002226 IFB11: CMP (R1),UOADR ;IF THIS ADR
4003 006732 001003 BNE IFC11 ;EQUALS UNIT #0 ADR
4004 006734 062701 000004 ADD #4,R1 ;INCREMENT TEMP #1 BY 4
4005 006740 000420 BR EIA11 ;BR TO END IF 'A'
4006 006742 005737 002230 IFC11: TST U1ADR ;IF UN1 ADDRESS
4007 006746 001005 BNE IFD11 ;NOT LOADED PREVIOUSLY
4008 006750 012137 002230 MOV (R1)+,U1ADR ;LOAD UNIT#1 ADR
4009 006754 012137 002304 MOV (R1)+,U1VECT ;LOAD UNIT #1 VECTOR
4010 006760 000405 BR EIC11 ;BR TO END IF 'C'
4011 006762 021137 002230 IFD11: CMP (R1),U1ADR ;IF UNIT ADR
4012 006766 001155 BNE ELD11 ;EQUALS UNIT #1 ADR
4013 006770 062701 000004 ADD #4,R1 ;THEN ADD 4 TO TEMP #1
4014 006774 012737 000001 007354 EIC11: MOV #1,UNT ;SET UNT=1
4015 007002 000240 EIA11: NOP ;
4016 007004 005737 002152 IFE11: TST RXXX ;IF RXXX
4017 007010 001446 BEQ IFI11 ;THEN
4018 007012 000240 IFF11: NOP ;
4019 007014 005711 TST (R1) ;IF DRIVE #0
4020 007016 001021 BNE IFH11 ;THEN
4021 007020 062701 000002 IFG11: ADD #2,R1 ;ADD 2 TO TEMP #1
4022 007024 005711 TST (R1) ;IF SIDE #0 SELECTED
4023 007026 001006 BNE ELG11 ;THEN
4024 007030 052737 000001 002212 BIS #1,SUT ;SET SIDE #0, DRIVE #0
4025 007036 005037 007352 CLR UNTCOD ;CLEAR UNIT CODE
4026 007042 000501 BR EIF11 ;BR TO END IF 'F'
4027 007044 052737 000004 002212 ELG11: BIS #4,SUT ;SET SIDE #1, DRIVE #0
4028 007062 012737 000002 007352 MOV #2,UNTCOD ;SET UNIT CODE = 10
4029 007060 000472 BR EIF11 ;BR TO END IF 'F'
4030 007062 062701 000002 IFH11: ADD #2,R1 ;ADD 2 TO TEMP #1
4031 007066 005711 TST (R1) ;IF SIDE #0 SELECTED
4032 007070 001007 BNE ELH11 ;THEN
4033 007072 052737 000002 002212 BIS #2,SUT ;SET SIDE #0, DRIVE #1
4034 007100 012737 000001 007352 MOV #1,UNTCOD ;SET UNIT CODE = 11
4035 007106 000457 BR EIF11 ;BR TO END IF 'F'
4036 007110 052737 000010 002212 ELH11: BIS #10,SUT ;SET SIDE #1, DRIVE #1
4037 007116 012737 000003 007352 MOV #3,UNTCOD ;SET UNIT CODE = 11
4038 007124 000450 BR EIF11 ;BR TO END IF 'F'
4039 007126 062701 000002 IFI11: ADD #2,R1 ;ADD 2 TO R1
4040 007132 005711 TST (R1) ;IF SIDE
4041 007134 001061 BNE ELI11 ;EQUALS 0, THEN
4042 007136 162701 000002 IFJ11: SUB #2,R1 ;SUBTRACT 2 FROM TEMP #1
4043 007142 005711 TST (R1) ;IF DRIVE
4044 007144 001020 BNE IFL11 ;EQUALS ZERO, THEN
4045 007146 005737 007354 IFK11: TST UNT ;IF UNIT
4046 007152 001006 BNE ELK11 ;EQUALS ZERO

```

```

4047 007154 052737 000001 002212      BIS      #1,SUT      ;SET UNIT #0, DRIVE #0
4048 007162 005037 007352      CLR      UNTCOD   ;CLEAR UNIT CODE
4049 007166 000427      BR       EIF11    ;BR TO END IF 'F'
4050 007170 052737 000004 002212  ELK11:  BIS      #4,SUT      ;SET UNIT #1, DRIVE #0
4051 007176 012737 000002 007352      MOV      #2,UNTCOD ;SET UNIT CODE = 10
4052 007204 000420      BR       EIF11    ;BR TO END IF 'F'
4053 007206 005737 007354      IFL11:  TST      UNT      ;IF UNIT
4054 007212 001007      BNE      ELL11    ;EQUALS 0
4055 007214 052737 000002 002212      BIS      #2,SUT      ;SET UNIT #0, DRIVE #1
4056 007222 012737 000001 007352      MOV      #1,UNTCOD ;SET UNIT CODE = 01
4057 007230 000406      BR       EIF11    ;BR TO END IF 'F'
4058 007232 052737 000010 002212  ELL11:  BIS      #10,SUT     ;SET UNIT #1, DRIVE #1
4059 007240 012737 000003 007352      MOV      #3,UNTCOD ;SET UNIT CODE = 11
4060 007246 000240      EIF11:  NOP
4061 007250 012701 007356      MOV      #UTO0,R1  ;GET BEGINING OF UNIT CODE TABLE
4062 007254 013702 007352      MOV      UNTCOD,R2 ;GET UNIT CODE
4063 007260 006302      ASL      R2      ;DOUBLE R2 FOR ADDRESSING
4064 007262 060201      ADD      R2,R1    ;FIND ADDRESS FOR THIS UNIT CODE
4065 007264 013703 006620      MOV      UNIT,R3   ;GET LOGICAL UNIT#
4066 007270 062703 000001      ADD      #1,R3    ;ADD 1 TO LOGICAL UNIT# TO PRINT USER UNIT#
4067 007274 010311      MOV      R3,(R1)  ;SET USER UNIT# FOR PRINT OUT
4068 007276 000424      BR       END11    ;BR TO END MOD
4069 007300      EL111:  PRINTF  INMSG2    ;
4070 007320 000413      BR       END11    ;BR TO END MOD
4071 007322      ELD11:  PRINTF  INMSG3,UNIT ;
4072 007346 000240      NOP
4073 007350 000207      END11:  RTS      PC      ;RETURN
4074
-----
4075 007352 000000      UNTCOD:  0      ;UNIT CODE
4076 007354 000000      UNT:     0      ;UNIT FLAG
4077 007356 177777      UTO0:    -1     ;****UUT CODE# TABLE****
4078 007360 177777      UT01:    -1     ;>STORAGE OF USER UNIT#
4079 007362 177777      UT10:    -1     ;FOR PRINT OUT &
4080 007364 177777      UT11:    -1     ;LOOKUP
-----
4081
4082 007366 047045 046445 051525  INMSG2:  .ASCIZ  /%N%MUST SELECT RXXX TO SELECT SIDE #1/
4083 007434 047045 052445 044516  INMSG3:  .ASCIZ  /%N%UNIT#%D1%NOT SCHEDULED-TWO BUS ADDRESS ONLY/
4084 007514      .EVEN
4085      ;MOD 1.1 ----- END MODULE -----

```

4088
4089
4090
4091
4092
4093
4094
4095 007514
4096
4103
4104 007514
4105
4117
4118
4119
4120 007520

.SBTTL CLEANUP CODING SECTION

;++
; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
; AT THE END OF EACH PASS.
;--

BGNCLN

EXIT CLN

.EVEN

ENDCLN

```

4123 . SBTTL DROP UNIT SECTION
4124
4125 ;++
4126 ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4127 ; TO NO LONGER BE TESTED.
4128 ;--
4129
4130 007522 BGNDU
4131
4132 007522 010037 007704 MOV R0,UNITDP ;GET LOGICAL UNIT #
4133 007526 005002 CLR R2 ;LET R2=UNIT CODE# & UNIT COUNT /CLEAR IT!
4134 007530 062737 000001 007704 ADD #1,UNITDP ;MAKE LOGICAL UNIT# = USER UNIT#
4135 007536 012701 007356 MOV #UTOO,R1 ;GET BEGIN UNIT CODE ADDRESS
4136 007542 023721 007704 15: CMP UNITDP,(R1)+ ;IF USER UNIT#
4137 007546 001420 BEQ 25 ;IS = UNIT CODE - UNIT#
4138 007550 005202 INC R2 ;INCREMENT UNIT CODE# & UNIT COUNT
4139 007552 022702 000005 CMP #5,R2 ;IF MAX # OF UNITS
4140 007556 101371 BHI 15 ;EXCEEDED
4141 007560 000240 NOP ;THEN
4142 007562 PRINTB #DUMSG2,UNITDP ;PRINT UNIT# NOT FOUND
4143 007606 000434 BR 35 ;BR TO EXIT
4144 007610 010237 002526 25: MOV R2,CVUNIT ;SET UNIT CODE FOR CONVERSION
4145 007614 004737 002426 CALL CVUTST ;CALL MOD U.A. 1 CONVERT UNIT# TO SUT CODE
4146 007620 013737 002530 007706 MOV SUTCV,SUTDRP ;SET SUT DROP CODE = SUT CONVERTED CODE
4147 007626 000240 NOP ;
4148 007630 013737 002530 007706 MOV SUTCV,SUTDRP ;GET SUT CODE
4149 007636 043737 007706 002212 BIC SUTDRP,SUT ;DROP UNIT SPEC IN SUTDRP
4150 007644 043737 007706 002210 BIC SUTDRP,SDD ;CLEAR UNIT SPEC IN SUT DROP
4151 007652 PRINTB #DUMSG1,UNITDP ;
4152 007676 000240 NOP ;
4153
4154
4155
4156
4157
4158
4159 007700 35: EXIT DU
4160
4161 007704 000000 UNITDP: 0 ;UNIT TO BE DROPPED
4162 007706 000000 SUTDRP: 0 ;SYS UNDER TST, DROP BIT
4163
4164 007710 047045 040445 042040 DUMSG1: .ASCIZ /%N%A DROP UNIT#%D1%A FROM TEST%/
4165 007751 045 022516 020101 DUMSG2: .ASCIZ /%N%A COULD NOT DROP UNIT#%D1%A -NOT SELECTED%/
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180 . EVEN
4181
4182 010030 ENDDU
  
```

4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4200
4201
4202
4214
4215
4216
4217
4218

010032

010032

010036

.SBTTL ADD UNIT SECTION
; ++
; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
; TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING. IF
; "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
; --

BGNAU

EXIT AU

.EVEN

ENDAU

```

4220 . TITLE HARDWARE TESTS
4221 . SBTTL TEST 1:
4222 . ++
4223 . TEST TO EXERCISE RX02/03 SYSTEM
4224 . --
4225 010040 . BGNTST
4226 . SBTTL MOD 0.0 - EXERCISE A SYSTEM
4227 . -----
4228
4229 010040 000240 . CONTRL: NOP
4230 010042 005037 010244 . BDB00: CLR EXCMP ; CLEAR EXERCISE COMPLETE
4231 010046 005037 002206 . CLR ABORT ; CLEAR ABORT FLAG
4232 010052 012737 000001 010242 . MOV #1,INITL ; SET INITIALIZE FLAG
4233 010060 005037 002242 . CLR RETRY ; CLEAR RETRY FLAGS
4234 010064 005037 002210 . CLR SDD ; CLEAR SYS DRIVES DONE
4235 010070 005037 002232 . CLR SYSERR ; CLEAR SYSTEM ERROR FLAGS
4236 010074 005037 002234 . CLR ERRTP ; CLEAR DEVICE ERROR FLAGS
4237 010100 005037 002236 . CLR CSRUUT ; CLEAR UUT CSR
4238 010104 005037 002240 . CLR ESRUUT ; CLEAR UUT ESR
4239 010110 005037 002312 . CLR XERUUT ; CLEAR UUT TEST ERROR REG
4240 010114 012737 000001 015644 . MOV #1,SUTPTR ; PRESET SYS UNDER TST PTR
4241 010122 004737 010246 . CALL GTSYEX ; CALL MOD 1.0 GET SYS EXER.
4242 010126 005737 002206 . IFA00: TST ABORT ; IF ABORT
4243 010132 001404 . BEQ IFCOO ; NOT = 0, THEN
4244 010134 005737 002232 . IFB00: TST SYSERR ; IF SYS ERR
4245 010140 001012 . BNE THDOO ; EQUALS 0, THEN
4246 010142 000416 . BR THEOO ; BR TO THEN 'E'
4247 010144 005737 002232 . IFCOO: TST SYSERR ; IF SYS ERR
4248 010150 001006 . BNE THDOO ; NOT=0, THEN BR TO THEN 'D', ELSE
4249 010152 . BDA00: BGNSEG ; BEGIN SEGMENT FOR ERROR LOOPS
4250 010154 004737 015110 . CALL SCSYEX ; CALL MOD 2.0 - SCDED SYS EX
4251 010160 005737 002232 . IFDOO: TST SYSERR ; IF SYS ERR
4252 010164 001410 . BEQ ELDOO ; NOT=0, THEN
4253 010166 004737 026650 . THDOO: CALL OTSYER ; CALL MOD 4.0 - O/P SYS ERR
4254 010172 005737 002206 . IFEOO: TST ABORT ; IF ABORT
4255 010176 001403 . BEQ ELDOO ; NOT=0, THEN
4256 010200 000240 . THEOO: NOP ;
4257 010202 . DOCLN ; DO CLEAN UP
4258 010204 000414 . BR ENDOO ; BR TO END
4259 010206 . ELDOO: CKLOOP ; SEE IF LOOP ON ERROR
4260 010210 . ENDSEG ; END SEGMENT FOR ERROR LOOPS
4261 010212 004737 026626 . CALL OTEXCM ; CALL MOD 3.0 O/P SYS. EX. COMPLETE
4262 010216 005737 010244 . DUA00: TST EXCMP ; DO UNTIL EXERCISE COMPLETE
4263 010222 001753 . BEQ BDA00 ; EQUALS 1
4264 010224 032737 100000 002164 . DUB00: BIT #100000,SWREG ; IF LOOP SET
4265 010232 001303 . BNE BDB00 ; NOT SET, THEN
4266 010234 . DORPT
4267 010236 000240 . ENDOO: NOP
4268 010240 . ENDTST
4269 . -----
4270 010242 000000 . INITL: 0 ; INITIALIZE POINTERS FLAG
4271 010244 000000 . EXCMP: 0 ; EXERCISE COMPLETE FLAG
4272 . MOD 0.0 ----- END MODULE -----

```



```
4274  
4275  
4276  
4277  
4278 010246 000240  
4279 010250 004737 010312  
4280 010254 000240  
4281 010256 032737 040000 002164 IFA10: BIT #40000, SWREG ; IF NO INITIALIZE  
4282 010264 001002 BNE ELA10 ; NOT SET, THEN  
4283 010266 004737 010436 JSR PC, GTSYS  
4284 010272 000240 ELA10: NOP  
4285 010274 004737 013474 JSR PC, GTEX  
4286 010300 000240 NOP  
4287 010302 005037 010310 CLR FIRST ; CLEAR FIRST PASS FLAG  
4288 010306 000207 RTS PC  
4289  
4290 010310 000001 FIRST: 1 ; FIRST PASS FLAG  
4291 ; MOD 1.0 ----- END MODULE -----  
4292  
4293
```

4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325

SBTTL MOD 1.1 - GET EXERCISE CONDITIONS

```
-----  
GTEXCD: NOP  
IFA11: BIT #1, SWREG ; IF SET FOR DOUBLE DENSITY  
; THEN  
BEQ ELA11 ;  
MOV #200, WDCNT ; SET WORD COUNT=256 BYTES  
BR EIA11 ; BR TO END IF 'A'  
ELA11: MOV #100, WDCNT ; SET WORD COUNT=128 BYTES  
EIA11: MOV OTDITK, OD ; SET OUTSIDE TRACK ADR. (FROM SOFTW P-TAB)  
MOV INDITK, ID ; SET INSIDE TRACK ADR. (FROM SOFT P-TAB)  
NOP  
BIT #2, SWREG ; IF DEL DATA SET  
BEQ ELB11 ; THEN  
MOV #10, DELDAT ; SET DEL DATA MODE  
BR EIB11 ; BR TO END IF 'B'  
ELB11: CLR DELDAT ; CLEAR DEL DATA MODE  
EIB11: NOP ;  
IFC11: BIT #1, SWREG ; IF DOUBLE DEN IS SET IN SOFT SWREG  
BEQ ELC11 ; THEN  
MOV #400, DEN ; SET DEN=DOUBLE  
BR EIC11 ; BR TO END IF 'C'  
ELC11: CLR DEN ; SET DEN=SINGLE  
EIC11: NOP ;  
RTS PC ; RETURN  
; MOD 1.1 ----- END MODULE -----
```

```

4327
4328
4329
4330          .SBTTL  MOD 1.2 - GET SYSTEM TO EXERCISE
4331          ;-----
4332 010436      GTSYS:  BRESET          ;ISSUE BUS RESET
4333 010440      012737 000040 021550      MOV      #DNBIT, RDYWD      ;SET READY WORD = DONE
4334 010446      013737 002226 021552      MOV      UOADR, CSRADR     ;SET ADDRESS
4335 010454      004737 021450              CALL     DELAY             ;CALL MOD - DELAY FOR DONE
4336 010460      032777 000040 171540      IFA12:  BIT      #DNBIT, UOADR ;IF UNIT #0 DONE BIT
4337 010466      001011              BNE     ELA12             ;NOT SET THEN
4338 010470      013737 007356 012114      MOV      UTOO, UNITN      ;SET UNIT # FOR PRINT
4339 010476      012737 012116 012112      MOV      #INTER2, ITMSG   ;SET PRINT MSG#
4340 010504      004737 012062              CALL     ITPRNT           ;PRINT-UO-NO DONE BIT
4341 010510      000445              BR      EIA12            ;BR TO END IF 'A'
4342 010512      012777 040000 171506      ELA12:  MOV      #40000, UOADR ;ELSE-ISSUE PROG INIT TO UO
4343 010520      012737 000040 021550      MOV      #DNBIT, RDYWD     ;SET READY WORD = DONE
4344 010526      013737 002226 021552      MOV      UOADR, CSRADR     ;SET TEST ADDRESS
4345 010534      004737 021450              CALL     DELAY             ;CALL MOD - DELAY FOR DONE
4346 010540      032777 000040 171460      IFB12:  BIT      #DNBIT, UOADR ;IF UNIT #0 DONE BIT
4347 010546      001011              BNE     ELB12             ;NOT SET THEN
4348 010550      013737 007356 012114      MOV      UTOO, UNITN      ;SET UNIT# FOR PRINT
4349 010556      012737 012214 012112      MOV      #INTER3, ITMSG   ;SET PRINT MSG#
4350 010564      004737 012062              CALL     ITPRNT           ;PRINT-UO, NO DONE BIT
4351 010570      000415              BR      EIA12            ;BR TO END IF 'A'
4352 010572      012737 000002 011432      ELB12:  MOV      #2, UNTCNT  ;SET # DRVS TO CK
4353 010600      012737 000001 011436      MOV      #1, SUTPOS       ;SET POSITION IN SUT TO TEST = 1
4354 010606      005037 011434              CLR     UNTCO             ;SET UUT CODE = 0
4355 010612      013704 002226              MOV     UOADR, R4         ;SET TEMP #4 = UO ADDRESS
4356 010616      004737 011102              CALL     CKDVAU           ;CALL MOD 1.2.1 - CK DRIVE STATUS
4357 010622      000412              BR      IFC12            ;BR TO IF 'C'
4358 010624      005737 002152              EIA12:  TST      RXXX      ;IF RXXX
4359 010630      001404              IFH12:  BEQ     ELH12      ;THEN
4360 010632      042737 000017 002212      BIC     #17, SUT          ;CLEAR RXXX UO SELECTED DRIVES
4361 010640      000403              BR      IFC12            ;BR TO IF 'C'
4362 010642      042737 000003 002212      ELH12:  BIC     #3, SUT          ;CLEAR RX02 UO SELECTED DRIVES
4363 010650      005737 002152              IFC12:  TST      RXXX      ;IF RXXX
4364 010654      001401              BEQ     IFD12            ;THEN
4365 010656      000471              BR      IFG12            ;BR TO IF 'G'
4366 010660      032737 000014 002212      IFD12:  BIT      #14, SUT      ;IF U1
4367 010666      001465              BEQ     IFG12            ;SELECTED THEN
4368 010670      032777 000040 171332      IFE12:  BIT      #DNBIT, U1ADR ;IF U1 DONE BIT
4369 010676      001446              BEQ     ELE12            ;SET THEN
4370 010700      012777 040000 171322      MOV      #40000, U1ADR     ;INITIALIZE DEVICE U1
4371 010706      012737 000040 021550      MOV      #DNBIT, RDYWD     ;SET READY WORD = DONE BIT
4372 010714      013737 002230 021552      MOV      U1ADR, CSRADR     ;SET TEST ADR
4373 010722      004737 021450              CALL     DELAY             ;CALL MOD - WAIT FOR DONE
4374 010726      032777 000040 171274      IFF12:  BIT      #DNBIT, U1ADR ;IF U1 DONE BIT
4375 010734      001416              BEQ     ELF12            ;SET THEN
4376 010736      012737 000004 011436      MOV      #4, SUTPOS       ;SET POSITION IN SUT = 4
4377 010744      012737 000002 011432      MOV      #2, UNTCNT       ;SET # DRVS TO CK = 2
4378 010752      012737 000002 011434      MOV      #2, UNTCO        ;SET UUT CODE = 2
4379 010760      013704 002230              MOV     U1ADR, R4         ;SET TEMP #4 = U1 ADR
4380 010764      004737 011102              CALL     CKDVAU           ;CALL MOD 1.2.1 - CK DRIVE STATUS
4381 010770      000424              BR      IFG12            ;BR TO IF 'G'
4382 010772      013737 007362 012114      ELF12:  MOV      UT10, UNITN  ;SET UNIT# FOR PRINT
  
```



```

4398          SBTTL MOD 1.2.1 - CK DRIVE AVAILABLE
4399          ;-----
4400
4401 011102 012737 000001 011416 CKDVAV: MOV      #1, REFORM      ;SET REFORMAT CK
4402 011110 010437 011426          MOV      R4, ITCSAD      ;SAVE C & S ADR
4403 011114 062704 000002          ADD      #2, R4          ;SET DATA BUFFER ADR
4404 011120 010437 011430          MOV      R4, ITDBAD      ;SAVE DB ADR
4405 011124 000240          BDA121: NOP
4406 011126 033737 011436 002212 IFA121: BIT      SUTPOS, SUT      ; IF BIT IN TEMP #1 & SUT
4407 011134 001512          BEQ      EIA121          ; EQUAL, THEN
4408 011136 013701 011434          MOV      UNTCD, R1      ;SAVE UNIT #
4409 011142 006301          ASL      R1              ;DOUBLE UNIT CD FOR ADR
4410 011144 062701 007356          ADD      #UTOO, R1      ;FIND ADR UNITS
4411 011150 011137 012114          MOV      (R1), UNITN    ;SET UNITS FOR PRINT
4412 011154 032737 000001 011434 IFB121: BIT      #1, UNTCD      ; IF DRIVE #1 SET IN TEMP #3
4413 011162 001407          BEQ      ELB121          ; THEN
4414 011164 012737 000033 011422          MOV      #33, INTCMD    ;SET READ STATUS DRV #1
4415 011172 012737 000001 011424          MOV      #1, DRIVEN     ;SET PRINT FOR DRV #1
4416 011200 000405          BR       EIB121          ;BR TO END IF 'B'
4417 011202 012737 000013 011422 ELB121: MOV      #13, INTCMD   ;SET READ STATUS DRV #0
4418 011210 005037 011424          CLR      DRIVEN         ;SET PRINT FOR DRIVE #0
4419 011214 013777 011422 000204 EIB121: MOV      INTCMD, #ITCSAD ;EXECUTE READ STATUS ON DRIVE AT TEMP #4
4420 011222 013737 011426 021552          MOV      ITCSAD, CSRADR ;PASS DOWN ADRS
4421 011230 012737 000040 021550          MOV      #DNBIT, RDYWD  ;PASS DOWN "DONE" BIT TO TEST
4422 011236 004737 021450          CALL     DELAY           ;CALL MOD - DELAY FOR DONE BIT
4423 011242 032777 000010 000160 IFH121: BIT      #10, #ITDBAD ; IF AC LOW BIT
4424 011250 001404          BEQ      IFC121          ; SET, THEN
4425 011252 012737 013340 012112          MOV      #ITER3, ITMSG  ;SET MSG# - AC LOW
4426 011260 000426          BR       EIC121          ;BR TO END IF 'C'
4427 011262 032777 000200 000140 IFC121: BIT      #200, #ITDBAD ; IF DRV RDY BIT
4428 011270 001004          BNE      IFD121          ; NOT SET, THEN
4429 011272 012737 012343 012112          MOV      #ITMSG1, ITMSG ;SET MSG# - NO DRIVE READY
4430 011300 000416          BR       EIC121          ;BR TO END IF 'C'
4431 011302 005737 002152          IFD121: TST      RXXX     ; IF UNIT IS TO BE TESTED AS RXXX
4432 011306 001420          BEQ      IFG121          ; THEN
4433 011310 032737 000002 011434 IFE121: BIT      #2, UNTCD      ; IF SIDE #1
4434 011316 001414          BEQ      IFG121          ; SELECTED
4435 011320 032777 000002 000102 IFF121: BIT      #2, #ITDBAD   ; IF SIDE #1
4436 011326 001010          BNE      IFG121          ; NOT READY, THEN
4437 011330 012737 012402 012112          MOV      #ITMSG2, ITMSG ;SET MSG#
4438 011336 004737 012062          EIC121: CALL     ITPRNT    ;CALL PRINT - NO SIDE RDY
4439 011342 004737 013454          CALL     ITDROP         ;CALL DROP UNIT
4440 011346 000405          BR       EIA121          ;BR TO ENDIF 'A'
4441 011350 005737 011416          IFG121: TST      REFORM    ; IF REFORMAT FLAG
4442 011354 001402          BEQ      EIA121          ; NOT SET, THEN
4443 011356 004737 011442          CALL     REFDRV         ;CALL REFORMAT DRIVE DENSITY
4444 011362 006137 011436          EIA121: ROL      SUTPOS    ;MOVE SELECT BIT TO TEST SYS UNDER TEST
4445 011366 005337 011432          DEC      UNTCNT         ;DECREMENT UNIT COUNT
4446 011372 005237 011434          INC      UNTCD          ;INCREMENT UNIT UNDER TEST CODE
4447 011376 005737 011432          DUA121: TST      UNTCNT    ;DO
4448 011402 001402          BEQ      END121         ;UNTIL
4449 011404 000137 011124          JMP      BDA121         ;ALL UNITS DONE
4450 011410 005037 011416          END121: CLR      REFORM   ;CLEAR REFORMAT CK
4451 011414 000207          RTS      PC             ;RETURN
4452          ;-----

```

4454			;	-----	
4455	011416	000000	REFORM: 0		; REFORMAT CHECK
4456	011420	000000	REFCMD: 0		; REFORMAT COMMAND
4457	011422	000000	INTCMD: 0		; INITIAL COMMAND WORD
4458	011424	000000	DRIVEN: 0		; DRIVE NUMBER
4459	011426	000000	ITCSAD: 0		; INITIAL C & S ADR
4460	011430	000000	ITDBAD: 0		; INITIAL DATA BUFFER ADR
4461	011432	000000	UNTCNT: 0		; UNIT COUNT
4462	011434	000000	UNTCOD: 0		; UNIT CODE
4463	011436	000000	SUTPOS: 0		; SYS UNDER TST POSITION
4464	011440	000000	FORMCK: 0		; FORMATT CK FLAG
4465			;	-----	

```

4467          . SBTTL MOD 1.2.1.1 - REFORMAT DRIVE DENSITY
4468          ;-----
4469
4470 011442 033737 011436 002212 REFDRV: BIT      SUTPOS, SUT      ; IF UNIT SELECTED IN
4471 011450 001002                BNE      IA1211        ; SYS UNDER TEST
4472 011452 000137 012060                JMP      X1211        ; THEN
4473 011456 032737 000001 002164 IA1211: BIT      #1, SWREG      ; IF DOUBLE DENSITY
4474 011464 001412                BEQ      IC1211        ; SET, THEN
4475 011466 032777 000040 177734 IB1211: BIT      #40, @ITDBAD    ; IF DISKETTE IS DOUBLE DENSITY
4476 011474 001171                BNE      X1211        ; THEN BR TO END, ELSE
4477 011476 012737 012440 012112        MOV      @ITMSG3, ITMSG ; SET MSG# DSK SGL DEN
4478 011504 004737 012062                CALL     ITPRNT        ; CALL PRINT -
4479 011510 000414                BR       LB1211        ; BR TO ELSE 'B'
4480 011512 032777 000040 177710 IC1211: BIT      #40, @ITDBAD    ; IF DISKETTE
4481 011520 001557                BEQ      X1211        ; IS NOT SINGLE DENSITY, THEN
4482 011522 012737 012646 012112        MOV      @ITMSG6, ITMSG ; SET MSG# DSK DBL DEN
4483 011530 004737 012062                CALL     ITPRNT        ; CALL PRINT -
4484 011534 005037 011420                CLR      REFCMD        ; SET REFORMAT CMD TO SINGLE DENSITY
4485 011540 000403                BR       ID1211        ; BR TO IF 'D'
4486 011542 012737 000400 011420 LB1211: MOV      #BIT8, REFCMD ; SET REFORMAT CMD TO DOUBLE DENSITY
4487 011550                ID1211: MANUAL
4488 011552                BNCOMPLETE          LD1211 ; IF MANUAL INTERVENTION
4489 011554                GMANIL              FCKMSG, FORMCK, 1, YES ; IS ALLOWED, THEN
4490 011570 005737 011440                IE1211: TST      FORMCK    ; IF REFORMATT
4491 011574 001516                BEQ      LE1211        ; OK, THEN
4492 011576 005037 011440                CLR      FORMCK        ; CLEAR REFORMATT CK
4493 011602 052737 000011 011420        BIS      #11, REFCMD    ; SET REFORMAT CMD
4494 011610 032737 000001 011434 IF1211: BIT      #1, UNTC D    ; IF DRIVE #1
4495 011616 001403                BEQ      IG1211        ; SELECTED
4496 011620 052737 000020 011420        BIS      #BIT4, REFCMD ; SET DRIVE #1 ON REFORMAT CMD
4497 011626 005737 002152        IG1211: TST      RXXX      ; IF RXXX
4498 011632 001407                BEQ      EG1211        ; DEVICE AND
4499 011634 032737 000002 011434        BIT      #2, UNTC D    ; SIDE #1
4500 011642 001403                BEQ      EG1211        ; SELECTED, THEN
4501 011644 052737 001000 011420        BIS      #BIT9, REFCMD ; SET SIDE #1 ON REFORMAT CMD
4502 011652 013777 011420 177546 EG1211: MOV      REFCMD, @ITCSAD ; SEND REFORMAT CMD
4503 011660 013737 011426 021552        MOV      @ITCSAD, CSRADR ; PASS UNIT ADRS
4504 011666 012737 000200 021550        MOV      @TRBIT, RDYWD  ; PASS "TR" BIT TO TEST
4505 011674 004737 021450                CALL     DELAY          ; CALL DELAY
4506 011700 005737 002232        IH1211: TST      SYSERR     ; IF
4507 011704 001046                BNE      LH1211        ; T. O. ERR
4508 011706 012777 000111 177514        MOV      #111, @ITDBAD  ; SEND VARIFY WORD (ASCII "I")
4509 011714 012737 013067 012112        MOV      @ITMSG9, ITMSG ; SET MSG# WRG DEN REFORMAT
4510 011722 004737 012062                CALL     ITPRNT        ; CALL PRINT -
4511 011726 013737 011426 021552        MOV      @ITCSAD, CSRADR ; SET UNIT BUS ADR
4512 011734 012737 000040 021550        MOV      @DNBIT, RDYWD  ; SET DONE BIT TST
4513 011742 012737 001000 021544        MOV      #1000, RYDX    ; SET DELAY MULT HIGH
4514 011750 004737 021450                CALL     DELAY          ; DELAY UNTIL DONE OR T. O.
4515 011754 012737 000020 021544        MOV      #20, RYDX      ; RESET DELAY MULT
4516 011762 032777 000040 177436 I11211: BIT      #40, @ITCSAD    ; IF DONE BIT
4517 011770 001410                BEQ      L11211        ; SET, THEN
4518 011772 032777 100000 177426 IJ1211: BIT      #100000, @ITCSAD ; IF ERR BIT NOT SET
4519 012000 001427                BEQ      X1211        ; THEN BR TO EXIT
4520 012002 012737 013146 012112        MOV      @ITER1, ITMSG  ; ELSE, SET "ERROR ON REFORMATT" MSG
4521 012010 000417                BR       ED1211        ; BR TO END IF 'D'
4522 012012 012737 013244 012112 L11211: MOV      @ITER2, ITMSG  ; SET "NO DONE BIT AFTER REFORMATT" MSG

```

```
4523 012020 000413 BR ED1211 ;BR TO END IF 'D'  
4524 012022 012737 012526 012112 LH1211: MOV #ITMSG4,ITMSG ;SET MSG# NO "TR" BIT TIME OUT ERR  
4525 012030 000407 BR ED1211 ;BR TO END IF 'D'  
4526 012032 012737 012735 012112 LE1211: MOV #ITMSG7,ITMSG ;SET MSG# DISK WRG DEN  
4527 012040 000403 BR ED1211 ;BR TO END IF 'D'  
4528 012042 012737 013003 012112 LD1211: MOV #ITMSG8,ITMSG ;SET MSG# MAN INTERVENTION NOT ALL  
4529 012050 004737 012062 ED1211: CALL ITPRNT ;CALL PRINT - MSG SET UP  
4530 012054 004737 013454 CALL ITDROP ;CALL DROP UNIT  
4531 012060 000207 X1211: RTS PC ;RETURN  
4532 ;-----
```



```

4534
4535          .SBTTL  MOD U.1.2 - INITIALIZE PRINT
4536          ;-----
4537
4538 012062 000240  ITPRNT: NOP          ;
4539 012064          PRINTB  IMSG,UNITN ;
4540 012110 000207  RTS          PC          ;RETURN
4541          ;-----
4542 012112 000000  IMSG: 0          ;INITIALIZE MSG#
4543 012114 000000  UNITN: 0         ;UNIT# FOR PRINT
4544          ;-----
4545 012116 047045 040445 047125 INTER2: .ASCIZ  /%AUNIT%D1A---NO DONE BIT AFTER BUS INITIALIZE - DROP UNIT/
4546 012214 047045 040445 047125 INTER3: .ASCIZ  /%AUNIT%D1A---NO DONE BIT AFTER PROG. INITIALIZE - DROP UNIT/
4547 012314 047045 040445 047040 INTER4: .ASCIZ  /%A NO SYSTEM TO TEST/
4548 012343      045 022516 052501 IMSG1: .ASCIZ  /%AUNIT%D1A- NO DRIVE READY/
4549 012402 047045 040445 047125 IMSG2: .ASCIZ  /%AUNIT%D1A- NO SIDE READY/
4550 012440 047045 040445 047125 IMSG3: .ASCIZ  /%AUNIT%D1A- WRONG DENSITY-SINGLE DENSITY DISKETTE/
4551 012526 047045 040445 047125 IMSG4: .ASCIZ  /%AUNIT%D1A- NO "TR" BIT AFTER SET DENSITY CMD/
4552 012610 047045 040445 047125 IMSG5: .ASCIZ  /%AUNIT%D1A- WRONG DENSITY/
4553 012646 047045 040445 047125 IMSG6: .ASCIZ  /%AUNIT%D1A-WRONG DENSITY - DOUBLE DENSITY DISKETTE/
4554 012735      045 022516 052501 IMSG7: .ASCIZ  /%AUNIT%D1A DISKETTE WRONG DENSITY/
4555 013003      045 022516 020101 IMSG8: .ASCIZ  /%A UNIT%D1A MAN. INTERVENTION REQ'D - REFORMAT/
4556 013067      045 020101 047125 IMSG9: .ASCIZ  /%A UNIT%D1A-REFORMATTING, DO NOT INTERRUPT%/
4557 013146 047045 040445 047125 ITER1: .ASCIZ  /%AUNIT%D1A- ERROR BIT SET AFTER REFORMAT COMMAND SEQUENCE/
4558 013244 047045 040445 047125 ITER2: .ASCIZ  /%AUNIT%D1A- NO DONE BIT AFTER REFORMAT COMMAND SEQUENCE/
4559 013340 047045 040445 047125 ITER3: .ASCIZ  /%AUNIT%D1A- AC LOW BIT SET/
4560 013377      104 051511 042513 FCKMSG: .ASCIZ  /DISKETTE WILL BE REFORMATTED - ARE YOU SURE?/
4561          .EVEN
4562          ;-----
4563 013454 013737 011436 007706 ITDROP: MOV      SUTPOS,SUTDRP ;SETUP SYS. UNDER TEST DROP BIT
4564 013462          DODU      UNTC      ;DROP THIS UNIT FROM TEST
4565 013470 000240          NOP          ;
4566 013472 000207  RTS          PC          ;RETURN
4567          ;MOD 1.2.1 ----- END MODULE -----
4568

```

```

4570
4571
4572
4573
4574 013474 000240          GTEX:  NOP
4575 013476 000240          NOP
4576 013500 000240          NOP
4577 013502 013737 002160 014154  MOV    TSTPAT,PAT      ;GET TEST PATTERN #
4578 013510 000240          NOP
4579 013512 004737 013602          CALL  STSTPA          ;CALL MOD 1. 3. 1 SET TEST PATTERN
4580 013516 013737 002162 015062  MOV    TRKSEQ,SEQUEN ;GET TRACK SEQ #
4581 013524 000240          NOP
4582 013526 013737 002166 015056  MOV    OTDITK,OD      ;GET OUTSIDE DIA. TRK
4583 013534 013737 002170 015060  MOV    INDITK,ID      ;GET INSIDE DIA. TRK
4584 013542 004737 014156          CALL  STKSEQ          ;CALL MOD 1. 3. 2 SET TRACK SEQUENCE
4585 013546 005737 010310          IFB13: TST    FIRST    ;IF A FIRST PASS
4586 013552 001007          BNE   THC13          ;THEN
4587 013554 032737 000040 002164  IFC13: BIT    #40,SWREG ;IF CLEAR STATISTICAL TABLES
4588 013562 001406          BEQ   END13          ;IS SELECTED THEN
4589 013564 042737 000040 002164  BIC   #40,SWREG      ;CLEAR SELECTED - CLR STAT TABLE
4590 013572 004737 015064          THC13: CALL  CLRSTA    ;CALL MOD 1. 3. 3 - CLEAR STATISTICAL TABLES
4591 013576 000240          NOP
4592 013600 000207          END13: RTS    PC     ;RETURN
4593
4594
4595
;MOD 1. 3 ----- END MODULE -----

```

```
4597 . SBTTL MOD 1.3.1 - SET DATA PATTERN
4598 -----
4599 PAT # DATA PATTERN
4600 -----
4601 0 NO PATTERN SPECIFIED (FORCE RANDOM DATA)
4602 1 ALL ZEROS
4603 2 ALL ONES
4604 3 FLOATING ZERO
4605 4 FLOATING ONE
4606 5 ALTERNATING BITS
4607 6 ALTERNATING PAIRS OF BITS
4608 7 RANDOM
4609 ; NOTE: DATA PATTERNS WILL BE MODIFIED SO BYTE #0 WILL CONTAIN TRACK ADDRESS
4610 ; AND BYTE #1 THE SECTOR ADDRESS IN WHICH THE DATA IS WRITTEN.
4611 ; THE LAST TWO BYTES CONTAIN THE CHECK SUM NUMBERS.
4612 -----
4613 013602 042737 000377 013666 STSTPA: BIC #377, @#BRONPT ; CLEAR BRANCH OFFSET
4614 013610 005037 014150 CLR SUM ; SET UP FOR ACCUMULATION OF CHECK SUM
4615 013614 005737 014154 TST PAT ; IF NO PATTERN SPECIFIED FORCE PATTERN 7
4616 013620 001003 BNE 15
4617 013622 012737 000007 014154 MOV #7, PAT
4618 013630 013704 014154 15: MOV PAT, R4 ; GET PATTERN BITS
4619 013634 005304 DEC R4 ; ADJUST FOR CORRECT OFFSET
4620 013636 006304 ASL R4
4621 013640 150437 013666 BISB R4, @#BRONPT ; INSERT OFFSET
4622 013644 012704 035564 MOV #DATPAT+2, R4 ; SET UP ADDRESS OF FIRST BYTE
4623 013650 013705 002270 MOV WDCNT, R5 ; SETUP WORD COUNT
4624 013654 006305 ASL R5 ; DOUBLE WORD COUNT FOR ADR
4625 013656 062705 035562 ADD #DATPAT, R5 ; ADD DATA PATTERN ADR
4626 013662 162705 000004 SUB #4, R5 ; ADJ. FOR CHECKSUM
4627 013666 000777 BRONPT: BR ; BRANCH BY OFFSET SELECTED
4628 013670 000137 013724 JMP DATA0 ; 000 DATA BYTE
4629 013674 000137 013742 JMP DATA1 ; 377 DATA BYTE
4630 013700 000137 013752 JMP FLOAT0 ; FLOAT A 0 THROUGH ALL 1'S
4631 013704 000137 014020 JMP FLOAT1 ; FLOAT A 1 THROUGH ALL 0'S
4632 013710 000137 014026 JMP PAT125 ; 125/052 DATA WORD
4633 013714 000137 014052 JMP PAT333 ; 314/063 DATA WORD
4634 013720 000137 014062 JMP RANDAT ; RANDOM DATA BYTE
4635 -----
4636 ; LOAD SOFTWARE BUFFER WITH ALL ZEROS (P = 1)
4637 -----
4638 013724 005037 014152 DATA: CLR DATBYT
4639 013730 004737 014110 PG: JSR PC, LOAD ; GO LOAD THE DATA BUFFER
4640 013734 005705 TST R5 ; IF R5
4641 013736 001463 BEQ END131 ; NOT =0, THEN
4642 013740 000773 BR PG
4643 -----
4644 ; LOAD SOFTWARE BUFFER WITH ALL ONES (P = 2)
4645 -----
4646 013742 112737 000377 014152 DATA1: MOVB #377, DATBYT
4647 013750 000767 BR PG
```

```

4649
4650 ;-----
4651 ;FLOAT A 0 THROUGH ONES IN SOFTWARE BUFFER (P = 3)
4652 013752 112737 000376 014152 FLOATO: MOVB #376,DATBYT ;SET UP A ONES FIELD
4653 013760 000261 XPG: SEC ;SET THE C BIT TO ROTATE THROUGH THE DATA
4654 013762 012702 000000 15: MOV #0,R2 ;CLR R2 (CAN'T USE "CLR" AS IT CLEARS "C" BIT)
4655 013766 103001 BCC 25 ;BR IF THE "C" BIT IS CLEARED
4656 013770 005202 INC R2 ;SET R2 IF NOT
4657 013772 004737 014110 25: JSR PC,LOAD ;GO LOAD THE DATA BUFFER
4658 013776 005705 TST R5 ;IF R5
4659 014000 001442 BEQ END131 ;NOT ZERO THEN
4660 014002 000241 CLC
4661 014004 005702 TST R2 ;IS R2 NONZERO
4662 014006 001401 BEQ 35
4663 014010 000261 SEC ;YES, SET THE "C" BIT
4664 014012 106137 014152 35: ROLB DATBYT
4665 014016 000761 BR 15
4666 ;-----
4667 ;FLOAT A 1 THROUGH ALL ZEROS IN SOFTWARE BUFFER (P = 4)
4668
4669 014020 005037 014152 FLOAT1: CLR DATBYT
4670 014024 000755 BR XPG
4671 ;-----
4672 ;ALTERNATING 1 & 0 IN ONE BYTE AND COMPLIMENT INTO THE NEXT (P = 5)
4673
4674 014026 112737 000125 014152 PAT125: MOVB #125,DATBYT
4675 014034 004737 014110 XXPG: JSR PC,LOAD
4676 014040 005705 TST R5 ;IF R5
4677 014042 001421 BEQ END131 ;NOT ZERO THEN
4678 014044 105137 014152 COMB DATBYT
4679 014050 000771 BR XXPG
4680 ;-----
4681 ;ALTERNATING 0 AND 1'S PAIR IN ONE BYTE & COMPLIMENT INTO NEXT (P = 6)
4682
4683 014052 112737 000333 014152 PAT333: MOVB #333,DATBYT
4684 014060 000765 BR XXPG
4685 ;-----
4686 ;LOAD SOFTWARE BUFFER WITH RANDOM DATA PATTERN (P = 0 OR 7)
4687
4688 014062 004737 002332 RANDAT: JSR PC,RANGEN ;GET RANDOM NUMBER
4689 014066 113737 002424 014152 MOVB RANUM,DATBYT
4690 014074 004737 014110 JSR PC,LOAD
4691 014100 005705 TST R5 ;IF R5
4692 014102 001401 BEQ END131 ;NOT ZERO THEN
4693 014104 000766 BR RANDAT
4694 ;-----
4695 014106 000207 END131: RTS PC ;RETURN.
4696 ;-----
4697

```

```
4699  
4700 ;-----  
4701 014110 063737 014152 014150 LOAD: ADD DATBYT,SUM ;ACCUMULATE THE PATTERN CHECK SUM  
4702 014116 113724 014152 MOV B DATBYT,(R4)+ ;LOAD THE DATA BUFFER  
4703 014122 020504 CMP R5,R4 ;HAVE 124 BYTES BEEN GENERATED  
4704 014124 001401 BEQ 15 ;IF YES, RETURN  
4705 014126 000407 BR ENLD ;IF NO, RETURN TO PATTERN GENERATOR  
4706 014130 113724 014150 15: MOV B SUM,(R4)+ ;PUT CHECKSUM INTO TABLE  
4707 014134 005137 014150 COM SUM ;COMPLIMENT CHECKSUM  
4708 014140 113714 014150 MOV B SUM,(R4) ;PUT COMP CHECK SUM INTO TABLE  
4709 014144 005005 CLR R5 ;CLEAR TEMP #5 - FLAG DONE MODULE  
4710 014146 000207 ENLD: RTS PC ;RETURN  
4711 ;-----  
4712 014150 000000 SUM: 0  
4713 014152 000000 DATBYT: 0  
4714 014154 000000 PAT: 0  
4715 ;MOD 1.3.1 ----- END MODULE -----
```

4717
 4718
 4719
 4720
 4721
 4722
 4723
 4724
 4725
 4726
 4727
 4728
 4729
 4730
 4731
 4732
 4733
 4734
 4735
 4736
 4737
 4738
 4739
 4740
 4741
 4742
 4743
 4744
 4745
 4746
 4747
 4748
 4749
 4750
 4751
 4752
 4753
 4754
 4755
 4756
 4757
 4758
 4759
 4760
 4761
 4762
 4763
 4764
 4765
 4766
 4767
 4768
 4769
 4770
 4771
 4772

SBTTL MOD 1.3.2 - SET TRACK SEQUENCE

SEQ #	SEQUENCE
0	NO SEQUENCE SPECIFIED (DEFAULT TO SEQ 7)
1	INCREMENT FROM OD TO ID
2	DECREMENT FROM ID TO OD
3	DO PREVIOUS 2 SEQUENCES
4	BOUNCE BETWEEN ID AND OD
5	DECREASING BOUNCE
6	STROBE BETWEEN OD AND DECREMENTING ID
7	RANDOM TRACK SELECTION

```

STKSEQ: CLR      TKTBPT      ;CLEAR TRK TBL PTR
         CLR      PRESTK     ;CLEAR PRESENT TRK
         CLR      TARGET     ;CLEAR TARGET TRK
         MOV      #177,PRESTK ;INIT PRESENT TRK TO HANDLE TRK #0
         MOV      OD,TARGET   ;INIT OD AS TARGET TRACK
         CLR      XID        ;INIT WORDING ID AND OD LOCATIONS
         MOV      ID,XID      ;SAVE INSIDE DIA. IN TEMP INSIDE DIA.
         CLR      XOD        ;CLEAR TEMP OUTSIDE DIA
         MOV      OD,XOD      ;SAVE OUTSIDE DIA. IN TEMP OUTSIDE DIA.
         MOV      XID,TRKCNT  ;SET UP NUMBER OF TRACK MOVEMENTS
         SUB      XOD,TRKCNT  ;
         INC      TRKCNT     ;INCREMENT # OF TRACKS
         BGE      GTTK       ;IF # OF TRACKS IS NEGATIVE, THEN
         MOV      #100000,SYSERR ;SET SYSTEM ERROR
         JMP      ENDTKS     ;EXIT
GTTK:   MOV      TRKSEQ,SEQUEN ;GET TRACK SEQUENCE #
         BICB    #377,@#BRONTK ;CLEAR OUT BRANCH OFFSET
         TST     SEQUEN      ;IF TRACK SEQUENCE
         BNE     1$         ;EQUALS ZERO, THEN
         MOV     #7,SEQUEN   ;FORCE SEQ #7-RANDOM
1$:     MOV     SEQUEN,R4    ;GET SEQUENCE BITS
         DEC     R4         ;ADJUST FOR CORRECT OFFSET
         ASL     R4         ;
BRONTK: BISB    R4,@#BRONTK ;THIS BR INST. IS MODIFIED SELECTED TRACK SEQUENCE
         BR     .          ;BRANCH TO SELECTED TRACK SEQUENCE
         JMP     SEQ1
         JMP     SEQ2
         JMP     SEQ3
         JMP     SEQ4      ;BOUNCE ID TO OD
         JMP     SEQ5      ;DECREASING BOUNCE
         JMP     SEQ6      ;STROBE
         JMP     SEQ7      ;RANDOM
    
```

; INCREMENT FROM OD TO ID & RETURN TO OD

```

SEQ1:  CMPB    XID,PRESTK   ;IF PRESENT TRACK=ID
         BNE    1$         ;THEN
         MOV    #-1,TARGET  ;TERMINATE TABLE
         BR    2$         ;END SEQ1
1$:    MOV     XOD,TARGET   ;ELSE SET NEW TRACK-OUTSIDE DIA
         INC    XOD        ;INCREMENT OUTSIDE DIA
2$:    BR     NEWTRK      ;END SEQ1
    
```

```

4773 ;-----
4774 ; DECREMENT FROM ID TO OD
4775
4776 014424 123737 015052 015050 SEQ2:  CMPB  XOD,PRESTK  ;
4777 014432 001004          BNE  15          ;
4778 014434 012737 177777 015046          MOV  #-1,TARGET ; TERMINATE TABLE
4779 014442 000405          BR   25          ; END SEQ2
4780 014444 013737 015054 015046 15:  MOV  XID,TARGET  ; SET NEXT TRACK=INSIDIA
4781 014452 005337 015054          DEC  XID          ; DECREMENT INSIDE DIA
4782 014456 000547          25:  BR   NEWTRK
4783 ;-----
4784 ; INCREMENT THEN DECREMENT TRACKS
4785
4786 014460 005701          SEQ3:  TST  R1          ; IF MODE
4787 014462 001402          BEQ  15          ; NOT EQUAL TO ZERO
4788 014464 005001          CLR  R1          ; THEN CHANGE MODE
4789 014466 000756          BR   SEQ2        ; DO SEQ2
4790 014470 012701 000001 15:  MOV  #1,R1       ; ELSE CHANGE MODE
4791 014474 000735          BR   SEQ1        ; DO SEQ1
4792 ;-----
4793 ; BOUNCE BETWEEN ID & OD ONLY
4794
4795 014476 005701          SEQ4:  TST  R1          ; IF MODE
4796 014500 001405          BEQ  15          ; NOT EQUAL TO ZERO
4797 014502 113737 015052 015046          MOVB XOD,TARGET  ; THEN SET NEXT TRACK=OUTSIDE DIA
4798 014510 005001          CLR  R1          ; CHANGE MODE
4799 014512 000405          BR   25          ; BR
4800 014514 113737 015054 015046 15:  MOVB XID,TARGET  ; ELSE SET NEXT TRACK=INSIDE DIA
4801 014522 012701 000001          MOV  #1,R1       ; TERMINATE TABLE
4802 014526 005337 015044          25:  DEC  TRKCNT   ;
4803 014532 001003          BNE  35          ;
4804 014534 012737 177777 015046          MOV  #-1,TARGET  ; TERMINATE TABLE
4805 014542 000515          35:  BR   NEWTRK
4806 ;-----
4807 ; BOUNCE BETWEEN DECREASING ID AND INCREASING OD
4808
4809 014544 123737 015054 015052 SEQ5:  CMPB  XID,XOD     ; IF INSIDE & OUTSIDE DIA
4810 014552 001421          BEQ  25          ; NOT EQUAL
4811 014554 005701          TST  R1          ; THEN, IF MODE
4812 014556 001407          BEQ  15          ;
4813 014560 005001          CLR  R1          ; CHANGE MODE
4814 014562 013737 015052 015046          MOV  XOD,TARGET  ; SET NEXT TRACK=OUTSIDE DIA
4815 014570 005237 015052          INC  XOD         ; INCREMENT OUTSIDE DIA
4816 014574 000413          BR   35          ; END SEQ5
4817 014576 012701 000001 15:  MOV  #1,R1       ; CHANGE MODE
4818 014602 013737 015054 015046          MOV  XID,TARGET  ; SET NEXT TRACK=INSIDE DIA
4819 014610 005337 015054          DEC  XID         ; DECREMENT INSIDE DIA
4820 014614 000403          BR   35          ; END SET5
4821 014616 012737 177777 015046 25:  MOV  #-1,TARGET  ; TERMINATE TABLE
4822 014624 000464          35:  BR   NEWTRK

```

```

4824 ;-----
4825 ;STROBE BETWEEN OD AND DECREASING ID
4826
4827 014626 123737 015054 015052 SEQ6:  CMPB  XID,XOD
4828 014634 001416                BEQ    15
4829 014636 123737 015050 015052        CMPB  PRESTK,XOD      ; IF O. D. JUST DONE
4830 014644 001006                BNE    35            ; THEN
4831 014646 113737 015054 015046        MOVB  XID,TARGET     ; SET TO DO I. D.
4832 014654 005337 015054                DEC   XID            ; DECREMENT I. D. FOR NEXT
4833 014660 000407                BR    25            ;
4834 014662 113737 015052 015046 35:    MOVB  XOD,TARGET     ; ELSE SET TO DO O. D.
4835 014670 000403                BR    25
4836 014672 012737 177777 015046 15:    MOV   #-1,TARGET
4837 014700 000436                25:   BR    NEWTRK
4838 ;-----
4839 ;RANDOM SEQUENCING OF TRACKS - THE OD/ID LIMITS SHOULD BE SET > THAN HALF THE TRACKS.
4840
4841 014702 000240                SEQ7:  NOP
4842 014704 004737 002332                JSR   PC,RANGEN     ; GET A RANDOM NUMBER
4843 014710 042737 177600 002424        BIC   #177600,RANUM ; CLEAR ALL BUT LOW 7 BITS
4844 014716 123737 002424 015054 IDCOMP: CMPB  RANUM,XID     ; IF RANUM LARGER THAN ID ADDRESS
4845 014724 003401                BLE   ODCOMP        ; THEN
4846 014726 000765                BR    SEQ7          ; BR TO GET ANOTHER RANDOM NUMBER
4847 014730 123737 002424 015052 ODCOMP: CMPB  RANUM,XOD     ; IF RANUM SMALLER THAN OD ADDRESS
4848 014736 002001                BGE   PRESCK        ; THEN
4849 014740 000760                BR    SEQ7          ; BR TO GET ANOTHER RANDOM NUMBER
4850 014742 123737 002424 015050 PRESCK: CMPB  RANUM,PRESTK ; IF RANUM EQUALS PRESENT TRACK
4851 014750 001754                BEQ   SEQ7          ; GET ANOTHER RANDOM NUMBER
4852 014752 013737 002424 015046        MOV   RANUM,TARGET  ; RANUM OK PUT IT IN TARGET TRACK
4853 014760 005337 015044                DEC   TRKCNT
4854 014764 001003                BNE   15
4855 014766 012737 177777 015046        MOV   #-1,TARGET   ; TERMINATE TABLE
4856 014774 000400                15:   BR    NEWTRK
4857 ;-----
4858 014776 012702 035327        NEWTRK: MOV   #TRKTBL-1,R2
4859 015002 005237 015042                INC   TKTBPT
4860 015006 063702 015042                ADD   TKTBPT,R2
4861 015012 113712 015046                MOVB  TARGET,(R2)
4862 015016 005737 015046                TST   TARGET
4863 015022 100406                BMI   ENDTKS
4864 015024 000240                NOP
4865 015026 113737 015046 015050        MOVB  TARGET,PRESTK
4866 015034 000137 014332                JMP   BRONTK
4867 015040 000207        ENDTKS: RTS    PC
4868 ;-----
4869 015042 000000                TKTBPT: 0            ; TRACK TABLE POINTER
4870 015044 000000                TRKCNT: 0
4871 015046 000000                TARGET: 0
4872 015050 000000                PRESTK: 0
4873 015052 000000                XOD:    0
4874 015054 000000                XID:    0
4875 015056 000000                OD:     0
4876 015060 000000                ID:     0
4877 015062 000000                SEQUEN: 0
4878 ;MOD 1.3.2 ----- END MODULE -----
4879

```


4881
4882
4883
4884
4885 015064 000240
4886 015066 012701 004442
4887 015072 012702 006326
4888 015076 005021
4889 015100 020102
4890 015102 001375
4891 015104 000240
4892 015106 000207
4893
4894

SBTTL MOD 1.3.3 - CLEAR STATISTICAL TABLES

CLRSTA: NOP ;
MOV #CKSML,R1 ; SET UP BEGINNING ADDRESS
MOV #ENDST,R2 ; SET UP TABLE LENGTH
BDA133: CLR (R1)+ ; CLEAR ADDRESSED LOCATION
CMP R1,R2 ;
BNE BDA133 ; DO UNTIL LAST ADDRESS DONE
NOP ;
END133: RTS PC ; RETURN
;MOD 1.3.3 ----- END MODULE -----

```

4896
4897
4898
4899
4900
4901 015110 000240          SCSYEX: NOP
4902 015112 005737 010242  IFK20: TST      INITL
4903 015116 001417          BEQ      ELK20
4904 015120 012737 000001 020454  MOV     @1, INITK
4905 015126 005037 015646          CLR     EXHCP
4906 015132 005037 015650          CLR     BTHDRV
4907 015136 005037 015652          CLR     BDVSCD
4908 015142 005037 015660          CLR     DVDNCK
4909 015146 005037 015662          CLR     DRVDN
4910 015152 005037 015670          CLR     ERTSAV
4911 015156 005037 015664          ELK20: CLR     SFERR
4912 015162 033737 015644 002212  IFA20: BIT     SUTPTR, SUT
4913 015170 001406          BEQ     ELA20
4914 015172 004737 002532          CALL   CVSTUT
4915 015176 013737 002600 002214  MOV     UNITST, UUT
4916 015204 000410          BR     BDB20
4917 015206 006337 015644          ELA20: ASL     SUTPTR
4918 015212 022737 000020 015644  DUC20: CMP     @20, SUTPTR
4919 015220 003360          BGT     IFA20
4920 015222 000137 015622          JMP     EDC20
4921 015226          BDB20: BGNSEG
4922 015230 013737 002156 015642  MOV     TSTN, EXN
4923 015236 004737 015672          CALL   GETTST
4924 015242 013737 016364 015640  MOV     TSTWD, TST
4925 015250 032737 000400 015640  IFB20: BIT     @400, TST
4926 015256 001514          BEQ     ELB20
4927 015260 004737 026534          CALL   STDVDN
4928 015264 032737 004000 015640  IFC20: BIT     @4000, TST
4929 015272 001001          BNE    IF120
4930 015274 000411          BR     EIC20
4931 015276 023727 015650 000003  IF120: CMP     BTHDRV, #3
4932 015304 001066          BNE    IFL20
4933 015306 013737 015650 015662  MOV     BTHDRV, DRVDN
4934 015314 005037 015650          CLR     BTHDRV
4935 015320 013737 002214 015666  EIC20: MOV     UUT, RESTK
4936 015326 062737 002000 015666  BIS     @2000, RESTK
4937 015334 032737 001000 015640  IFF20: BIT     @1000, TST
4938 015342 001001          BNE    ELF20
4939 015344 000410          BR     EIF20
4940 015346 012737 002000 015656  ELF20: MOV     @2000, ADVTRK
4941 015354 005737 015646          IFG20: TST     EXHCP
4942 015360 001420          BEQ    IFH20
4943 015362 005037 015646          CLR     EXHCP
4944 015366 063737 015662 002210  EIF20: BIS     DRVDN, SDD
4945 015374 006337 015644          ASL     SUTPTR
4946 015400 013737 002214 015666  MOV     UUT, RESTK
4947 015406 062737 002000 015666  BIS     @2000, RESTK
4948 015414 005037 015650          CLR     BTHDRV
4949 015420 000506          BR     END20
4950 015422 005737 002224          IFH20: TST     DELDAT
4951 015426 001403          BEQ     ELH20
; IF INITIALIZE
; THEN
; SET INITIALIZE TRK FLG
; CLEAR EX HALF COMPL
; CLEAR BOTH DRV DONE FLG
; CLEAR BOTH DRV SEC DONE FLG
; CLEAR DRV DONE CK FLG
; CLEAR DRV DONE
; CLEAR ERR TYP SAVE
; CLEAR SFT ERR
; IF SYSTEM UNDER TEST BIT
; IS SET
; CALL MOD U.A. 2 - CONVERT SUTPTR-->UUT
; SET UNIT UNDER TEST
; BR TO BEGIN 'B'
; SHIFT SUT POINTER TO TEST
; DO UNTIL SUT POINTER
; EQUALS 10000 BIN
; BR TO END DO 'C'
; BEGIN SEGMENT FOR ERROR LOOPS
; GET TEST # = EXERCISE #
; CALL MOD 2.1 - GET A TEST
; SAVE TEST WORD
; IF NEXT UNIT BIT
; IS SET THEN
; CALL MOD 2.6 -SET DRIVES DONE
; IF ADV TRK BIT
; IS NOT SET THEN
; BR TO END IF 'C'
; IF BOTH DRIVES DONE
; THEN
; SET BOTH DRVS DONE TEST
; CLEAR BOTH DRIVES DONE FLAG & THEN
; SET UUT TO RESET TRK
; SET INC TRK ONTO RESET TRK
; IF DEL DATA CK BIT
; IS SET THEN
; BR TO IF 'F'
; SET ADV TRK = INCR TRK
; IF EXERCISE 1/2 COMPLETE
; IS SET, THEN
; CLEAR EX HALF COMPLETE
; SET THIS DRV DONE
; SETUP PTR TO CK NXT UNIT
; GET UUT
; SET INCTRK ON RESET TRK FLAG
; CLEAR BOTH DRV DN FLAG
; BR TO END
; IF DEL DATA MODE
; IS SET
    
```

```

4952 015430 005037 002224          CLR  DELDAT          ;CLEAR DEL DATA MODE
4953 015434 000403          BR    EIM20          ;BR TO END IF 'H'
4954 015436 012737 000010 002224  ELH20: MOV  #10,DELDAT  ;SET DEC DATA MODE
4955 015444 005037 015662          EIM20: CLR  DRVDM          ;CLEAR DRV DONE
4956 015450 012737 000001 015646  MOV  #1,EXHCP        ;SET EX 1/2 COMPLETE
4957 015456 000444          BR    EIB20          ;BR TO END IF 'B'
4958 015460 032737 000003 015652  IFL20: BIT  #3,BDVSCD  ;IF BOTH DRV SEC DONE
4959 015466 001405          BEQ  ELL20          ;THEN
4960 015470 005037 015652          CLR  BDVSCD         ;CLEAR DRV SEC DONE FLAGS
4961 015474 012737 004000 015656  MOV  #4000,ADVTRK   ;ALLOW TRACK ADVANCE
4962 015502 004737 016526          ELL20: CALL GTDRV          ;CALL MOD 2.2 - GET A DRIVE
4963 015506 000430          BR    EIB20          ;BR TO END IF 'B'
4964 015510 053737 015656 015654  ELB20: BIS  ADVTRK,INCTRK ;SET ADV TRK (IF SET BY PREV OP)
4965 015516 013737 015640 017520  MOV  TST,DRVST      ;PASS DRIVE TEST
4966 015524 004737 016714          CALL XDTVST         ;CALL MOD 2.3 - EXECUTE DRIVE TEST
4967 015530 000240          NOP                  ;
4968 015532 013737 017520 021634  MOV  DRVST,TSTEV    ;PASS DRIVE TEST FOR EVAL
4969 015540 004737 021554          CALL EVTSTR         ;CALL MOD 2.4 - EVAL. TEST RESULTS
4970 015544 013701 015640          MOV  TST,R1         ;GET DRV TST
4971 015550 042701 171777          BIC  #171777,R1     ;SAVE TRK BITS
4972 015554 010137 015654          MOV  R1,INCTRK     ;SET TRK BITS
4973 015560 005037 015656          CLR  ADVTRK        ;CLEAR ADV TRK FLAG
4974 015564 005037 010242          CLR  INITL         ;CLEAR INITIALIZE FLAG
4975 015570 000240          EIB20: NOP          ;
4976 015572 005737 002234          IFM20: TST  ERRTP   ;IF ERR TYPE
4977 015576 001402          BEQ  EIM20         ;NOT=0
4978 015600 004737 025106          CALL OTERTP        ;CALL MOD 2.5 - O/P ERR TYPE
4979 015604 000240          EIM20: NOP          ;
4980 015606 005737 002232          DUB20: TST  SYSERR  ;DO UNLESS SYSTEM ERROR
4981 015612 001011          BNE  END20         ;NOT=0 THEN
4982 015614          ENDSEG            ;END SEGMENT FOR ERROR LOOPS
4983 015616 000137 015226          JMP  BDB20         ;BR TO END MOD
4984 015622 012737 000001 015644  EDC20: MOV  #1,SUTPTR  ;SET SYS UNDER TEST PTR
4985 015630 052737 000001 015664  BIS  #1,SFERR      ;SET SFT ERR
4986 015636 000207          END20: RTS  PC     ;END MODULE
4987
-----
4988 015640 000000          TST:  0            ;TEST FOR EXECUTION
4989 015642 000000          EXN:  0            ;EXERCISE #
4990 015644 000001          SUTPTR: 1          ;SYSTEM UNDER TEST POINTER
4991 015646 000000          EXHCP: 0          ;EXERCISE HALF COMPLETE (EX#7) DEL DATA PASS
4992 015650 000000          BTHDRV: 0         ;BOTH DRIVES DONE FLAG
4993 015652 000000          BDVSCD: 0        ;BOTH DRIVE SECTORS DONE FLAG
4994 015654 000000          INCTRK: 0        ;INCREMENT TRACK FLAGS
4995 015656 000000          ADVTRK: 0        ;ADVANCE TRACK FLAG
4996 015660 000000          DVDNCK: 0        ;DRV DONE CK FLAG
4997 015662 000000          DRVDM: 0         ;DRIVE DONE
4998 015664 000000          SFERR: 0         ;SOFTWARE ERR
4999 015666 000000          RESTK: 0        ;RESET TRK FLAG
5000 015670 000000          ERTSAV: 0        ;ERR TYP SAVE REG
5001          ;MOD 2.0 ----- END MODULE -----
5002
    
```

```

5005          . SBTTL MOD 2.1 - GET A TEST
5006          ;-----
5007
5008 015672 000240          GETTST: NOP          ;
5009 015674 013701 015642  MOV          EXN,R1      ; GET EXERCISE NUMBER
5010 015700 006301          ASL          R1          ; DOUBLE EXERCISE NUMBER
5011 015702 012702 016372  MOV          #EXADTB,R2   ; GET EXERCISE ADDRESS TABLE
5012 015706 060102          ADD          R1,R2       ; CAL EXERCISE TO BE USED
5013 015710 011237 016362  MOV          (R2),EXADR   ; GET BEGIN ADR EXERCISE
5014 015714 005737 010242  IFL21: TST          INITL  ; IF INITIALIZE
5015 015720 001405          BEQ          IFA21       ; IS SET, THEN
5016 015722 005037 016360  CLR          TSTPTR      ; CLEAR TST PTR
5017 015726          IFF21: INLOOP        ; IF IN LOOP
5018 015730          BNCOMPLETE IFA21 ; SET, THEN
5019 015732 000576          BR          EIF21       ; BR TO END IF 'F'
5020 015734 005737 002242  IFA21: TST          RETRY  ; IF RETRY
5021 015740 001404          BEQ          IFB21      ; NOT=0, AND
5022 015742 032737 000004 002164  BIT          #BIT02,SWREG ; IF RETRY ON ERROR
5023 015750 001103          BNE          IFH21      ; IS NOT SET, THEN
5024 015752 005737 016360  IFB21: TST          TSTPTR ; IF TST PTR
5025 015756 001006          BNE          IFC21      ; EQUALS ZERO
5026 015760 012737 000002 016360  MOV          #2,TSTPTR   ; ADV. TST PTR 1 CMD
5027 015766 005037 016366  CLR          TBPTR       ; CLEAR TABLE PAIR COUNT
5028 015772 000556          BR          EIF21       ; BR TO END IF 'F'
5029 015774 005737 002300  IFC21: TST          SECDN  ; IF SECTOR DONE IS
5030 016000 001450          BEQ          IFG21      ; SET THEN
5031 016002 005737 016366  IFK21: TST          TBPTR  ; IF TABLE PAIR CNT=1,
5032 016006 001445          BEQ          IFG21      ; THEN
5033 016010 062737 000002 016360  ADD          #2,TSTPTR   ; ADVANCE ONE TEST CMD
5034 016016 005037 016366  CLR          TBPTR       ; CLEAR TABLE PAIR COUNT
5035 016022 005037 015660  CLR          DVDNCK      ; CLEAR DRV DONE CK FLAG
5036 016026 032737 040000 016364  IFD21: BIT          #40000,TSTWD ; IF DONE CK
5037 016034 001411          BEQ          ELD21      ; IS SET, THEN
5038 016036 005737 002276  TST          TRKDN      ; IF TRACK DONE IS
5039 016042 001406          BEQ          ELD21      ; SET, THEN
5040 016044 005037 002276  CLR          TRKDN      ; CLEAR TRK DONE
5041 016050 012737 000001 015660  MOV          #1,DVDNCK   ; SET DRV DONE CK
5042 016056 000524          BR          EIF21       ; BR TO END IF 'F'
5043 016060 000240          ELD21: NOP          ;
5044 016062 032737 006000 016364  IFM21: BIT          #6000,TSTWD ; IF ADV OR INCR TRK
5045 016070 001517          BEQ          EIF21      ; IS SET, THEN
5046 016072 032737 100000 016364  IFN21: BIT          #100000,TSTWD ; IF '4 CMD SEQ'
5047 016100 001404          BEQ          ELN21      ; IS SET, THEN
5048 016102 162737 000010 016360  SUB          #10,TSTPTR  ; BACK UP 4 CMDS
5049 016110 000507          BR          EIF21       ; BR TO END IF 'F'
5050 016112 162737 000004 016360  ELN21: SUB          #4,TSTPTR ; BACK UP TWO TEST CMDS
5051 016120 000503          BR          EIF21       ; BR TO END IF 'F'
5052 016122 005737 016366  IFG21: TST          TBPTR  ; IF TABLE PAIR COUNT
5053 016126 001406          BEQ          ELG21      ; EQUALS 1 THEN
5054 016130 005037 016366  CLR          TBPTR       ; CLEAR TABLE PAIR COUNT
5055 016134 162737 000002 016360  SUB          #2,TSTPTR   ; BACK UP ONE CMD
5056 016142 000472          BR          EIF21       ; BR END IF 'F'
5057 016144 005237 016366  ELG21: INC          TBPTR  ; INCREMENT TABLE PAIR COUNT
5058 016150 062737 000002 016360  ADD          #2,TSTPTR   ; ADVANCE ONE CMD
5059 016156 000464          BR          EIF21       ; BR END IF 'F'
5060 016160 032737 000010 002242  IFH21: BIT          #10,RETRY ; IF NO DATA RETRY IS
  
```

```

5061 016166 001005          BNE    IF121          ;SET, OR
5062 016170 032737 000020 002242 BIT    #20,RETRY     ;IF NO CRC RETRY IS
5063 016176 001001          BNE    IF121          ;SET, THEN
5064 016200 000453          BR     EIF21          ;BR END IF 'F'
5065 016202 032737 000002 002242 IF121: BIT    #2,RETRY     ;IF WRITE RETRY IS
5066 016210 001412          BEQ    IFJ21          ;SET, THEN
5067 016212 162737 000006 016360 SUB    #6,TSTPTR     ;BACK UP 3 CMDS
5068 016220 042737 000002 002242 BIC    #2,RETRY     ;CLEAR WRITE RETRY
5069 016226 012737 000003 016370 MOV    #3,TSVCT      ;SET TEST ADV COUNT=3
5070 016234 000433          BR     E1121          ;BR TO END IF 'I'
5071 016236 032737 000004 002242 IFJ21: BIT    #4,RETRY     ;IF READ RETRY IS
5072 016244 001412          BEQ    ELJ21          ;SET THEN
5073 016246 162737 000002 016360 SUB    #2,TSTPTR     ;BACK UP 1 CMD
5074 016254 042737 000004 002242 BIC    #4,RETRY     ;CLEAR READ RETRY
5075 016262 012737 000001 016370 MOV    #1,TSVCT      ;SET TEST ADV COUNT=1
5076 016270 000415          BR     E1121          ;BR TO END IF 'I'
5077 016272 005337 016370          ELJ21: DEC    TSVCT      ;DECREMENT TEST ADV COUNT
5078 016276 062737 000002 016360 ADD    #2,TSTPTR     ;ADV TEST POINTER 1 CMD
5079 016304 005737 016370          IF021: TST    TSVCT      ;IF TEST ADV COUNTER
5080 016310 001007          BNE    EIF21          ;EQUALS ZERO, THEN
5081 016312 005037 002242          CLR    RETRY         ;CLEAR RETRY
5082 016316 005237 016366          INC    TBRPCT        ;SET TABLE PAIR COUNT
5083 016322 000402          BR     EIF21          ;BR TO END IF 'F'
5084 016324 005037 016366          E1121: CLR    TBRPCT        ;CLEAR TABLE PAIR CNT
5085 016330 013703 016360          EIF21: MOV    TSTPTR,R3 ;GET TEST POINTER
5086 016334 063703 016362          ADD    EXADR,R3     ;CAL. CUR. TEST OF THIS EXERCISE
5087 016340 011337 016364          MOV    (R3),TSTWD   ;PASS UP TEST WORD
5088 016344 105713          IFE21: TSTB   (R3)   ;IF CMD LOWER BYTE
5089 016346 002002          BGE    EIE21          ;EQUALS -1, THEN
5090 016350 005037 016360          CLR    TSTPTR       ;RESET TEST PTR
5091 016354 000240          EIE21: NOP           ;
5092 016356 000207          RTS    PC            ;RETURN
5093                                     ;-----
5094 016360 000000          TSTPTR: .WORD 0      ;TEST POINTER
5095 016362 000000          EXADR:  .WORD 0      ;CURRENT EXERCISE TABLE BASE ADDRESS
5096 016364 000000          TSTWD:  .WORD 0      ;TEST WORD TO PASS UP
5097 016366 000000          TBRPCT: .WORD 0      ;TABLE PAIR COUNT
5098 016370 000000          TSVCT:  .WORD 0      ;TEST ADVANCE COUNTER
5099                                     ;-----
5100 016372 016506          EXADTB: .WORD EX7    ;EXERCISE ADDRESS TABLE
5101 016374 016412          .WORD  EX1
5102 016376 016422          .WORD  EX2
5103 016400 016436          .WORD  EX3
5104 016402 016452          .WORD  EX4
5105 016404 016462          .WORD  EX5
5106 016406 016472          .WORD  EX6
5107 016410 016506          .WORD  EX7
5108                                     ;-----
5109

```

```

5111 ;-----
5112 016412 177777 EX1: .WORD -1 ;-, / FILL BUFFER
5113 016414 000000 .WORD 0 ;-, / WRITE SECTOR
5114 016416 044002 .WORD 44002 ;DCK, ADVTRK /
5115 016420 000777 .WORD 777 ;NXTUNT, / -1
5116 016422 177777 EX2: .WORD -1 ;-, / FILL BUFFER
5117 016424 000000 .WORD 0 ;-, / WRITE SECTOR
5118 016426 000002 .WORD 2 ;-, / READ SECTOR
5119 016430 000003 .WORD 3 ;-, / READ SECTOR
5120 016432 154001 .WORD 154001 ;4CMD, DCK, ADVTRK, RAW, / EMPTY BUFFER
5121 016434 000777 .WORD 777 ;NXTUNT, / -1
5122 016436 177777 EX3: .WORD -1 ;-, / FILL BUFFER
5123 016440 000000 .WORD 0 ;-, / WRITE SECTOR
5124 016442 000002 .WORD 2 ;-, / READ SECTOR
5125 016444 000003 .WORD 3 ;-, / READ SECTOR
5126 016446 174001 .WORD 174001 ;4CMD, DCK, ADVTRK, DACK, RAW / EMPTY BUFFER
5127 016450 000777 .WORD 777 ;NXTUNT, / -1
5128 016452 177777 EX4: .WORD -1 ;-, / FILL BUFFER
5129 016454 000003 .WORD 3 ;-, / READ SECTOR
5130 016456 064001 .WORD 64001 ;DCK, ADVTRK, DATAACK, / EMPTY BUFFER
5131 016460 000777 .WORD 777 ;NXTUNT, / -1
5132 016462 177777 EX5: .WORD -1 ;-, / FILL BUFFER
5133 016464 000003 .WORD 3 ;-, / READ SECTOR
5134 016466 044001 .WORD 44001 ;DCK, ADVTRK, / EMPTY BUFFER
5135 016470 000777 .WORD 777 ;NXTUNT, / -1
5136 016472 177777 EX6: .WORD -1 ;-, / FILL BUFFER
5137 016474 000000 .WORD 0 ;-, / WRITE SECTOR
5138 016476 000002 .WORD 2 ;-, / WRITE SECTOR
5139 016500 000003 .WORD 3 ;-, / READ SECTOR
5140 016502 170001 .WORD 170001 ;4CMD, DCK, DATAACK, RAW, / EMPTY BUFFER
5141 016504 004777 .WORD 4777 ;ADVTRK, NXTUNT, / -1
5142 016506 177777 EX7: .WORD -1 ;-, / FILL BUFFER
5143 016510 000000 .WORD 0 ;-, / WRITE SECTOR
5144 016512 000002 .WORD 2 ;-, / WRITE SECTOR
5145 016514 000003 .WORD 3 ;-, / READ SECTOR
5146 016516 172001 .WORD 172001 ;4CMD, DCK, DACK, RAW, INCTK / EMPTY BUFFER
5147 016520 000003 .WORD 3 ;-, / READ SECTOR
5148 016522 064001 .WORD 64001 ;DCK, DATAACK, ADVTRK, / EMPTY BUFFER
5149 016524 001777 .WORD 1777 ;DDCHK, NXTUNT, / -1

```

```

5150 ;-----
5151 BIT# NUMONIC FUNCTION
5152 ---- -
5153 15 4CMD 4 COMMAND SEQUENCE
5154 14 DCK DONE CHECK
5155 13 DATAACK DO DATA CHECK
5156 12 RAW READ AFTER WRITE FLAG
5157 11 ADVTRK ADVANCE TRACK MODE
5158 10 INCTK INCREMENT TRACK MODE
5159 09 DDCHK DEL. DATA CHECK
5160 08 NXTUNT GET NEXT UNIT, IF DONE LAST UNIT
5161 ;
5162 ;MOD 2.1 ----- END MODULE -----
5163
5164

```

```
5166  
5167  
5168  
5169  
5170  
5171 016526 000240 GTDRV: NOP ;  
5172 016530 032737 000001 002214 IFA22: BIT #1,UUT ; IF UUT=DRIVE 0  
5173 016536 001024 BNE IFD22 ; THEN  
5174 016540 032737 000002 002214 IFB22: BIT #2,UUT ; IF UNIT/SIDE UNDER TEST (UUT)  
5175 016546 001404 BEQ ELB22 ; EQUALS 1  
5176 016550 012737 000010 016712 MOV #10,TSTSUT ; SET TEST OF SYS. UNDER TEST UNIT/SIDE=1  
5177 016556 000403 BR IFC22 ; BR TO IF 'C'  
5178 016560 012737 000002 016712 ELB22: MOV #2,TSTSUT ; SET TEST OF SYS. UNDER TEST UNIT/SIDE=0  
5179 016566 033737 016712 002212 IFC22: BIT TSTSUT,SUT ; IF DRIVE 1 SELECTED FOR TEST  
5180 016574 001404 BEQ ELC22 ; THEN  
5181 016576 052737 000001 002214 BIS #1,UUT ; SET UNIT UNDER TEST TO DRV #1  
5182 016604 000430 BR EIE22 ; BR TO END IF 'E'  
5183 016606 000417 ELC22: BR THE22 ; BR TO THEN 'E'  
5184 016610 032737 000002 002214 IFD22: BIT #2,UUT ; IF UNIT/SIDE UNDER TEST (UUT)  
5185 016616 001404 BEQ ELD22 ; EQUALS 1  
5186 016620 012737 000004 016712 MOV #4,TSTSUT ; SET TEST OF SYS. UNDER TEST UNIT/SIDE 1  
5187 016626 000403 BR IFE22 ; BR TO IF 'E'  
5188 016630 012737 000001 016712 ELD22: MOV #1,TSTSUT ; SET TEST OF SYS. UNDER TEST UNIT/SIDE 0  
5189 016636 033737 016712 002212 IFE22: BIT TSTSUT,SUT ; IF DRIVE 0 SELECTED FOR TEST  
5190 016644 001404 BEQ ELE22 ; THEN  
5191 016646 042737 000001 002214 THE22: BIC #1,UUT ; SET UNIT UNDER TEST TO DRV#0  
5192 016654 000404 BR EIE22 ; BR TO END IF 'E'  
5193 016656 052737 000001 002214 ELE22: BIS #1,UUT ; SET UNIT UNDER TEST TO DRV#1  
5194 016664 000240 NOP ;  
5195 016666 013704 002214 EIE22: MOV UUT,R4 ; GET UNIT UNDER TEST  
5196 016672 006304 ASL R4 ; DOUBLE IT  
5197 016674 010437 002220 MOV R4,UUTOFF ; SET UUT OFFSET  
5198 016700 062704 007356 ADD #UTOO,R4 ; GET UUT UNIT # FOR PRINT  
5199 016704 011437 006620 MOV (R4),UNIT ; SET UNIT=PRINT UNIT #  
5200 016710 000207 END22: RTS PC ; RETURN  
5201  
5202 016712 000000 TSTSUT: 0  
5203 ;MOD 2.2 ----- END MODULE -----  
5204
```

```

5206
5207
5208           SBTTL MOD 2.3 - EXECUTE DRIVE TEST
5209           ;-----
5210
5211 016714 013737 002270 017522 XDVTST: MOV      WDCNT,WDCT      ;SET DRIVE WORD CNT
5212 016722 013702 002220          MOV      UTOFF,R2      ;GET UUT OFFSET
5213 016726 005737 002152          IFA23: TST      RXXX        ;IF DEVICE IS AN
5214 016732 001010          BNE      1$          ;RX02 THEN
5215 016734 032737 000002 002214          BIT      #2,UUT      ;IF UNIT UNDER TEST IS
5216 016742 001404          BEQ      1$          ;#1 THEN
5217 016744 013737 002230 002216          MOV      U1ADR,UUTADR ;GET UNIT #1 UNIBUS ADR
5218 016752 000403          BR       EIA23       ;BR TO END IF 'A'
5219 016754 013737 002226 002216 1$: MOV      UOADR,UUTADR ;GET UNIT #0 UNIBUS ADR
5220 016762 000240          EIA23: NOP
5221 016764 005737 015666          IFI23: TST      RESTK      ;IF RESET TRK
5222 016770 001413          BEQ      IFB23       ;IF SET, THEN
5223 016772 113705 015666          MOV8     RESTK,R5      ;GET UUT OFFSET
5224 016776 006305          ASL     R5            ;DOUBLE OFFSET
5225 017000 062705 017502          ADD     #CTRK,R5      ;ADD TRK TABLE ADR
5226 017004 013715 002166          MOV     OTDITK,(R5)   ;RESET TO MIN TRK
5227 017010 005037 002300          CLR     SECDN        ;CLEAR SEC DONE FLAG
5228 017014 005037 015666          CLR     RESTK        ;CLEAR RESET TRK FLAG
5229 017020 005737 010242          IFB23: TST      INITL      ;IF INITIALIZE IS
5230 017024 001415          BEQ     EIB23        ;SET, THEN
5231 017026 012705 017472          MOV     #CSEC,R5     ;GET START OF CUR TRK & SEC TBL
5232 017032 012704 000004          MOV     #4,R4        ;SET TBL LENGTH
5233 017036 005025          1$: CLR     (R5)+      ;CLEAR TABLES
5234 017040 005304          DEC     R4            ;DECR TBL LENGTH
5235 017042 001375          BNE     1$           ;DO UNTIL LENGHT=0
5236 017044 012704 000004          MOV     #4,R4        ;SET TBL LENGTH
5237 017050 013725 002166          2$: MOV     OTDITK,(R5)+ ;SET STARTING TRACKS
5238 017054 005304          DEC     R4            ;DECREMENT TBL LENGTH
5239 017056 001374          BNE     2$           ;DO UNTIL LENGTH=0
5240 017060 012701 017502          EIB23: MOV     #CTRK,R1 ;GET BEGIN ADR DRIVE CURRENT TRK.
5241 017064 060201          ADD     R2,R1        ;CAL. DRIVE CUR. TRK. LOCATOR
5242 017066 010137 017514          MOV     R1,CNTKLC    ;SAVE DRV. CUR. TRK.
5243 017072 017737 000416 020442          MOV     @CNTKLC,CURTRK ;GET DRIVE CUR. TRK.
5244 017100 012701 017472          MOV     #CSEC,R1     ;GET BEGIN ADR DRIVE CUR. SEC.
5245 017104 060201          ADD     R2,R1        ;CAL. DRIVE CUR. SEC. LOCATOR
5246 017106 010137 017512          MOV     R1,CNSCLC    ;SAVE DRV CUR SEC LOC.
5247 017112 017737 000374 020106          MOV     @CNSCLC,CURSEC ;GET DRIVE CUR SEC.
5248 017120          IFJ23: INLOOP      ;IF IN LOOP
5249 017122          BNCOMPLETE IFC23    ;THEN
5250 017124 000526          BR      EIJ23        ;BR TO END IF 'I'
5251 017126 005737 002242          IFC23: TST      RETRY     ;IF RETRY IS
5252 017132 001443          BEQ     IFG23        ;NOT=0, AND
5253 017134 032737 000004 002164          BIT     #BIT02,SWREG ;IF RETRY ON ERR
5254 017142 001437          BEQ     IFG23        ;IS SET, THEN
5255 017144 032737 000001 002242          IFD23: BIT     #1,RETRY  ;IF SEEK RETRY
5256 017152 001001          BNE     1$           ;IS = 0
5257 017154 000404          BR      2$           ;THEN BR TO 2$
5258 017156 032737 000010 002164 1$: BIT     #BIT03,SWREG ;ELSE IF RECAL SWITCH
5259 017164 001003          BNE     THD23        ;IS NOT SET
5260 017166 005037 017516          2$: CLR     SEEK      ;THEN CLEAR SEEK FUNCTION FLAG
5261 017172 000420          BR      EID23        ;BR TO END IF 'D'

```



```

5262 017174 012737 040000 020610 THD23: MOV #40000,DVTST ;PASS PROGRAM INITIALIZE TO DRIVE TEST
5263 017202 004737 020456 CALL GTDVFN ;CALL MOD 2.3.3 GET DRIVE FUNCTION
5264 017206 013737 017526 017524 MOV DRVFN,WDOT ;PASS DRIVE FUNCTION
5265 017214 013737 002216 021252 MOV UUTADR,CSADR ;SET ADR FOR DRIVE FUNCTION
5266 017222 004737 020612 CALL OTDVFN ;CALL MOD 2.3.4 O/P DRIVE FUNCTION
5267 017226 012737 000001 017516 MOV #1,SEEK ;SET SEEK FLAG
5268 017234 005037 002242 EID23: CLR RETRY ;CLEAR RETRY FLAGS
5269 017240 000460 BR EIJ23 ;BR TO END IF 'C'
5270 017242 013705 017520 IFG23: MOV DRVTST,R5 ;SETUP DRIVE TST
5271 017246 042705 177770 BIC #177770,R5 ;FOR TYPE CK
5272 017252 005705 TST R5 ;IF DRIVE TST
5273 017254 001404 BEQ IFE23 ;IS NOT 'FILL BUFF'
5274 017256 022705 000003 CMP #3,R5 ;OR
5275 017262 001401 BEQ IFE23 ;NOT 'READ SEC' , THEN
5276 017264 000434 BR IFH23 ;BR TO IF 'H'
5277 017266 005737 002300 IFE23: TST SECDN ;IF SEC DONE
5278 017272 001417 BEQ ELE23 ;IS = 1
5279 017274 005737 015654 IFF23: TST INCTRK ;IF INCR TRK FLAGS
5280 017300 001414 BEQ ELE23 ;ARE SET , THEN
5281 017302 013737 015654 020440 MOV INCTRK,TRKINC ;PASS TRK FLAGS
5282 017310 004737 020142 CALL GETTRK ;CALL MOD 2.3.2 GET TRACK
5283 017314 013777 020442 000172 MOV CURTRK,@CNTKLC ;SAVE CURRENT TRACK
5284 017322 012737 000001 017516 MOV #1,SEEK ;SET SEEK FLAG
5285 017330 000402 BR EIE23 ;BR TO END IF 'E'
5286 017332 005037 017516 ELE23: CLR SEEK ;RESET SEEK
5287 017336 017737 000150 020106 EIE23: MOV @CNSCLC,CURSEC ;PASS CURRENT SECTOR
5288 017344 004737 017530 CALL GETSEC ;CALL MOD 2.3.1 GET A SECTOR
5289 017350 013777 020106 000134 MOV CURSEC,@CNSCLC ;SAVE UPDATED CURRENT SECTOR
5290 017356 032737 000006 017520 IFH23: BIT #6,DRVTST ;IF DRIVE TST
5291 017364 001006 BNE EIJ23 ;IS 'FILL BUFF' , THEN
5292 017366 012701 035562 MOV #DATPAT,R1 ;SET UP DATA PATTERN ADR
5293 017372 117721 000116 MOVB @CNTKLC,(R1)+ ;SET TRK ADR IN DATA BUF BYTE #0
5294 017376 117711 000110 MOVB @CNSCLC,(R1) ;SET SEC ADR IN DATA BUF BYTE#1
5295 017402 005037 020610 EIJ23: CLR DVTST ;CLEAR DRIVE TEST
5296 017406 113737 017520 020610 MOVB DRVTST,DVTST ;PASS DRIVE TEST
5297 017414 004737 020456 CALL GTDVFN ;CALL MOD 2.3.3 GET DRIVE FUNCTION
5298 017420 013737 017526 017524 MOV DRVFN,WDOT ;PASS FUNCTION WORD (PASS TO 2.3.4)
5299 017426 017737 000062 021254 MOV @CNTKLC,TRKADR ;PASS CURRENT TRACK (PASS TO 2.3.4)
5300 017434 017737 000052 021256 MOV @CNSCLC,SECADR ;PASS CURRENT SECTOR (PASS TO 2.3.4)
5301 017442 013737 002216 021252 MOV UUTADR,CSADR ;PASS UUT C&S ADR (PASS TO 2.3.4)
5302 017450 004737 020612 CALL OTDVFN ;CALL MOD 2.3.4 O/P DRIVE FUNCTION
5303 017454 013737 021254 002272 MOV TRKADR,TRACK ;SAVE TRACK ADR IN GLOBAL
5304 017462 013737 021256 002274 MOV SECADR,SECTOR ;SAVE SECTOR ADR IN GLOBAL
5305 017470 000207 RTS PC ;RETURN
5306

```

```
5308  
5309 ;-----  
5310 017472 000000 CSEC: . WORD 0 ; CURRENT DRV SECTOR TABLE  
5311 017474 000000 . WORD 0  
5312 017476 000000 . WORD 0  
5313 017500 000000 . WORD 0  
5314 017502 000000 CTRK: . WORD 0 ; CURRENT DRV TRK TABLE  
5315 017504 000000 . WORD 0  
5316 017506 000000 . WORD 0  
5317 017510 000000 . WORD 0  
5318  
5319 017512 000000 CNSCLC: . WORD 0 ; CURRENT SECTOR LOCATOR  
5320 017514 000000 CNTKLC: . WORD 0 ; CURRENT TRACK LOCATOR  
5321 017516 000000 SEEK: . WORD 0 ; SEEK FLAG  
5322 017520 000000 DRVTST: . WORD 0 ; DRIVE TEST  
5323 017522 000000 WDCT: . WORD 0 ; WORD COUNT  
5324 017524 000000 WDOT: . WORD 0 ; FUNCTION WORD TO SEND OUT  
5325 017526 000000 DRVFN: . WORD 0 ; DRIVE FUNCTION WORD  
5326 ; MOD 2.3 ----- END MODULE -----  
5327
```

```

5329          . SBTTL MOD 2.3.1 - GET A SECTOR
5330          ;-----
5331
5332 017530 005037 020102 GETSEC: CLR      UTSCDN      ;CLEAR UUT SECTOR DONE
5333 017534 013705 002214          MOV      UUT,R5        ;GET UNIT UNDER TST
5334 017540 006305          ASL      R5            ;DOUBLE FOR WRD ADR
5335 017542 005737 010242 IFI231: TST      INITL      ;IF INITIALIZE IS
5336 017546 001406          BEQ      E11231       ;SET, THEN
5337 017550 012701 020062          MOV      #SSEC,R1     ;GET STARTING SEC ADR
5338 017554 005021          CLR      (R1)+        ;CLEAR UNTO0 SSEC
5339 017556 005021          CLR      (R1)+        ;CLEAR UNTO1 SSEC
5340 017560 005021          CLR      (R1)+        ;CLEAR UNT10 SSEC
5341 017562 005011          CLR      (R1)         ;CLEAR UNT11 SSEC
5342 017564 012701 020062 E11231: MOV      #SSEC,R1     ;GET START SECTOR BASE ADR
5343 017570 060501          ADD      R5,R1        ;FIND ADR UUT START SECTOR (TEMP 1)
5344 017572 011102          MOV      (R1),R2      ;SAVE UUT STARTING SECTOR (TEMP 2)
5345 017574 012703 020072          MOV      #NSEC,R3     ;GET NEXT SECTOR BASE ADR
5346 017600 060503          ADD      R5,R3        ;FIND ADR UUT NEXT SECTOR (TEMP 3)
5347 017602 011304          MOV      (R3),R4      ;SAVE UUT NEXT SECTOR (TEMP 4)
5348 017604 020237 002172 IFA231: CMP      R2,MINSEC   ;IF STARTING SECTOR < MIN. SECTOR
5349 017610 103422          BLO      ELA231       ;THEN
5350 017612 010437 020106          MOV      R4,CURSEC    ;SET CURRENT SECTOR=UUT NEXT SECTOR
5351 017616 023737 020056 020104 IFG231: CMP      SCPSCT,INTLV ;IF SECTOR PASS CNT< INTERLV
5352 017624 103053          BHIS     THF231       ;THEN BR TO THEN 'F',ELSE
5353 017626 005737 020060 IFH231: TST      STSCFG    ;IF START SEC FLAG
5354 017632 001405          BEQ      ELH231       ;IS SET, THEN
5355 017634 005037 020060          CLR      STSCFG       ;CLEAR FLAG
5356 017640 010204          MOV      R2,R4        ;SET DRV NXT SEC= DRV START SEC
5357 017642 010213          MOV      R2,(R3)      ;SAVE DRV NXT SEC
5358 017644 000426          BR       IFC231       ;BR TO IF 'C'
5359 017646 063704 020104 ELH231: ADD      INTLV,R4   ;NSEC=NSEC+INTERLV
5360 017652 010413          MOV      R4,(R3)      ;SAVE NEXT SEC
5361 017654 000422          BR       IFC231       ;BR TO IF 'C'
5362 017656 013737 002172 020106 ELA231: MOV      MINSEC,CURSEC ;SET CURRENT SECTOR = MIN. SECTOR
5363 017664 013711 002172          MOV      MINSEC,(R1)  ;SET UUT START SECTOR = MIN. SECTOR
5364 017670 013702 002172          MOV      MINSEC,R2    ;SET R2=MINSEC
5365 017674 005037 020056          CLR      SCPSCT       ;CLEAR SECTOR PASS COUNT
5366 017700 023737 002172 002174 IFB231: CMP      MINSEC,MAXSEC ;IF MAX. SECTOR NOT=MIN. SECTOR
5367 017706 001443          BEQ      ELB231       ;THEN
5368 017710 010205          THB231: MOV      R2,R5   ;GET UUT STARTING SECTOR
5369 017712 063705 020104          ADD      INTLV,R5     ;ADD SECTOR INTERLEAVE
5370 017716 010513          MOV      R5,(R3)      ;SAVE NEXT UUT NEXT SEC (TEMP 5)
5371 017720 010504          MOV      R5,R4        ;SAVE NEXT UUT NEXT SEC (TEMP 4)
5372 017722 020437 002174 IFC231: CMP      R4,MAXSEC   ;IF NEXT SECTOR > MAX. SECTOR
5373 017726 103432          BLO      ELC231       ;THEN
5374 017730 005211          INC      (R1)         ;INCREMENT UUT STARTING SECTOR
5375 017732 011102          MOV      (R1),R2      ;SET UP NEW START SEC
5376 017734 005237 020056          INC      SCPSCT       ;INCR SECTOR PASS CNT
5377 017740 020437 002174 IFD231: CMP      R4,MAXSEC   ;IF NXT SEC NOT = MAX SEC
5378 017744 001417          BEQ      ELD231       ;THEN
5379 017746 020237 002174 IFF231: CMP      R2,MAXSEC   ;IF DRV START SEC > MAX SEC
5380 017752 101411          BLOS     ELF231       ;THEN
5381 017754 012737 000001 020102 THF231: MOV      #1,UTSCDN    ;SET UUT SECTOR DONE
5382 017762 004737 020110          CALL     STSCDN       ;CALL MOD 2.3.1.A - SET DRIVE SECTOR DONE FLAG
5383 017766 005011          CLR      (R1)         ;CLEAR UUT STARTING SECTOR
5384 017770 005037 020056          CLR      SCPSCT       ;CLEAR SEC PASS CNT
    
```

```

5385 017774 000420          BR      END231      ;BRANCH TO END GET SECTOR
5386 017776 010213          ELF231: MOV     R2,(R3)    ;SET DRV NXT SEC = DRV START SEC
5387 020000 010204          MOV     R2,R4      ;SAVE DRV NXT SEC
5388 020002 000415          BR      END231      ;BR TO END
5389 020004 012737 000001 020060  ELD231: MOV     #1,STSCFG  ;SET START SEC FLAG
5390 020012 000411          BR      END231      ;BR TO END
5391 020014 000410          ELC231: BR      END231    ;BRANCH TO END GET SECTOR
5392 020016 012737 000001 020102  ELB231: MOV     #1,UTSCDN  ;SET DRIVE SECTOR DONE FLAG
5393 020024 004737 020110          CALL   STSCDN      ;CALL MOD 2.3.1.A - SET DRIVE SECTOR DONE FLAG
5394 020030 005037 020056          CLR    SCPSCT      ;CLEAR SEC PASS CNT
5395 020034 005011          CLR    (R1)        ;CLEAR UUT STARTING SECTOR
5396 020036 013737 020102 002300  END231: MOV     UTSCDN,SECDN
5397 020044 010437 020054          MOV     R4,NXSCSA
5398 020050 000240          NOP
5399 020052 000207          RTS     PC          ;RETURN TO MOD 2.3
5400          ;MOD 2.3.1 ----- REGISTERS & TABLES -----
5401 020054 000000  NXSCSA: 0
5402 020056 000000  SCPSCT: 0      ;SEC PASS COUNT
5403 020060 000000  STSCFG: 0      ;GET NEW STARTING SEC FLAG
5404 020062 000000  SSEC: 0        ;UUT STARTING SECTOR
5405 020064 000000          0
5406 020066 000000          0
5407 020070 000000          0
5408 020072 000000  NSEC: 0        ;UUT NEXT SECTOR
5409 020074 000000          0
5410 020076 000000          0
5411 020100 000000          0
5412 020102 000000  UTSCDN: 0      ;UUT SECTOR DONE FLAG
5413 020104 000003  INTLV: 3      ;SECTOR INTERLEAVE
5414 020106 000000  CURSEC: 0     ;CURRENT SECTOR UUT
5415          ;MOD 2.3.1 ----- END MODULE -----
5416
5417
5418
5419          ;SBTTL MOD 2.3.1.A - SET SECTOR DONE
5420          ;-----
5421
5422 020110 000240  STSCDN: NOP
5423 020112 032737 000001 002214  BIT     #1,UUT      ;IF DRIVE #1 DONE
5424 020120 001404          BEQ     15          ;THEN
5425 020122 052737 000002 015652  BIS     #2,BDVSCD   ;SET DRIVE #1 SEC DONE FLAG
5426 020130 000403          BR      25          ;BR TO END
5427 020132 052737 000001 015652  15:    BIS     #1,BDVSCD ;SET DRIVE #0 SEC DONE FLAG
5428 020140 000207  25:    RTS     PC          ;RETURN
5429          ;MOD 2.3.1.A ----- END MODULE -----

```

```
5431          ; SBTTL MOD 2.3.2 - GET A TRACK
5432          ; -----
5433
5434 020142 013737 002170 020436 GETTRK: MOV      INDITK,MAXTRK ;GET INSIDE DIA AS SET BY OP
5435 020150 013737 002166 020434          MOV      OTDITK,MINTRK ;GET OUTSIDE DIA AS SET BY OP
5436 020156 005737 020454          IFH232: TST      INITTK      ;IF INITIALIZE TRK IS
5437 020162 001413          BEQ      EIH232      ;SET, THEN
5438 020164 005037 020454          CLR      INITTK      ;RESET INITIALIZE TRK FLG
5439 020170 012701 020444          MOV      #TKTL,R1    ;GET START OF TRK TBL
5440 020174 005021          CLR      (R1)+       ;SET UNTOO
5441 020176 005021          CLR      (R1)+       ;SET UNTO1
5442 020200 005021          CLR      (R1)+       ;SET UNT10
5443 020202 005011          CLR      (R1)        ;SET UNT11
5444 020204 013737 020434 020442          MOV      MINTRK,CURTRK ;SET MIN CURRENT TRK
5445 020212 013702 002214          EIH232: MOV      UUT,R2  ;GET UNIT UNDER TEST INDICATOR
5446 020216 006302          ASL      R2          ;DOUBLE FOR ADDRESSING WORDS
5447 020220 005037 020432          CLR      TRKDNF      ;CLEAR TRACK DONE FLAG
5448 020224 032737 002000 020440          IFA232: BIT      #2000,TRKINC ;IF INCREMENT TRACK FLAG
5449 020232 001023          BNE      IFG232      ;NOT SET, THEN (USE SELECTED TRK SEQ)
5450 020234 012701 020444          MOV      #TKTL,R1    ;GET DRIVE TRACK TABLE LOCATOR BASE ADR
5451 020240 060201          ADD      R2,R1       ;CAL. DRV. TRK. TAB. LOCATOR ADR
5452 020242 011102          MOV      (R1),R2     ;GET DRV. TRK. TAB. LOCATOR
5453 020244 012703 035330          MOV      #TRKTBL,R3  ;GET BEGIN TRACK TABLE ADR
5454 020250 060203          ADD      R2,R3       ;CAL. TRACK TAB. ADR. THIS DRIVE
5455 020252 005202          INC      R2          ;INCREMENT DRV. TRK. TAB. LOCATOR
5456 020254 010211          MOV      R2,(R1)     ;SAVE DRV. TRK. TAB. LOCATOR
5457 020256 111337 020442          MOV      (R3),CURTRK ;SAVE CURRENT TRACK
5458 020262 005203          INC      R3          ;INCREMENT TRACK TAB. POINTER
5459 020264 105713          IFF232: TST      (R3)  ;IF NEXT TRACK
5460 020266 002004          BGE      ELF232      ;EQUALS -1
5461 020270 012737 000001 020432          MOV      #1,TRKDNF   ;THEN SET TRACK DONE FLAG
5462 020276 005011          CLR      (R1)        ;RESET DRV. TRK. TAB. LOCATOR ADR.
5463 020300 000445          ELF232: BR       END232  ;BR TO END MOD.
5464 020302 123737 020442 020436          IFG232: CMPB     CURTRK,MAXTRK ;IF CURRENT TRK > OR = MAX TRK (O. D.)
5465 020310 103403          BLO      IFB232      ;THEN
5466 020312 013737 020434 020442          MOV      MINTRK,CURTRK ;SET CURRENT TRK = MIN TRK
5467 020320 123737 020442 020434          IFB232: CMPB     CURTRK,MINTRK ;IF CURRENT TRK > OR = MIN TRK (O. D.)
5468 020326 103427          BLO      ELB232      ;THEN
5469 020330 013701 020442          MOV      CURTRK,R1   ;GET CURRENT TRACK
5470 020334 005201          INC      R1          ;INCREMENT CURRENT TRACK
5471 020336 120137 020436          IFC232: CMPB     R1,MAXTRK ;IF CURRENT TRK +1 < MAX TRK (I. D.)
5472 020342 103001          BHS      IFD232      ;THEN
5473 020344 000406          BR       EID232      ;BRANCH TO END IF 'D'
5474 020346 120137 020436          IFD232: CMPB     R1,MAXTRK ;IF CURRENT TRK +1 = MAX TRK
5475 020352 001006          BNE      IFE232      ;THEN
5476 020354 012737 000001 020432          MOV      #1,TRKDNF   ;SET TRK DONE FLAG
5477 020362 010137 020442          EID232: MOV      R1,CURTRK ;SAVE CURRENT TRK +1 = CURRENT TRK
5478 020366 000412          BR       END232      ;BR END OF MOD.
5479 020370 123737 020436 020434          IFE232: CMPB     MAXTRK,MINTRK ;IF TRK MAX = TRK MIN
5480 020376 001003          BNE      ELB232      ;THEN
5481 020400 012737 000001 020432          MOV      #1,TRKDNF   ;SET TRK DONE FLAG
5482 020406 013737 020434 020442          ELB232: MOV      MINTRK,CURTRK ;SET CURRENT TRK = MIN. TRK (O. D.)
5483 020414 013737 020432 002276          END232: MOV      TRKDNF,TRKDN ;SAVE TRACK DONE FLAG
5484 020422 000240          NOP
5485 020424 005037 020440          CLR      TRKINC      ;CLEAR TRK INCR FLAG
5486 020430 000207          RTS      PC
```

```
5487 ;-----  
5488 020432 000000 TRKDNF: . WORD 0 ; TRACK DONE FLAG  
5489 020434 000000 MINTRK: . WORD 0 ; MINIMUM TRACK - O. D.  
5490 020436 000000 MAXTRK: . WORD 0 ; MAXIMUM TRACK - I. D.  
5491 020440 000000 TRKINC: . WORD 0 ; INCREMENT TRK FLAG  
5492 020442 000000 CURTRK: . WORD 0 ; CURRENT TRACK  
5493 020444 000000 TKTL: . WORD 0 ; DRV TRK TABLE LOCATOR  
5494 020446 000000 . WORD 0  
5495 020450 000000 . WORD 0  
5496 020452 000000 . WORD 0  
5497 020454 000000 INITTK: . WORD 0 ; INITIALIZE TRK FLAG  
5498 ;MOD 2.3.2 ----- END MODULE -----
```

```

5500 . SBTTL MOD 2.3.3 - GET A DRIVE FUNCTION
5501 ;-----
5502
5503 020456 005001 GTDVFN: CLR R1 ;CLEAR REG #1
5504 020460 013701 020610 MOV DVTST,R1 ;GET DRIVE TEST
5505 020464 032701 040000 IFA233: BIT #40000,R1 ;IF NOT INITIALIZE
5506 020470 001012 BNE IFB233 ;THEN
5507 020472 042701 177700 BIC #177700,R1 ;CLEAR TOP BYTE OF R1
5508 020476 006301 ASL R1 ;FORMAT FUNCTION
5509 020500 052701 000001 BIS #1,R1 ;SET GO BIT
5510 020504 020127 000005 IFE233: CMP R1,#5 ;IF WRT FUNCT
5511 020510 001002 BNE IFB233 ;THEN
5512 020512 053701 002224 BIS DELDAT,R1 ;SET DEL DAT WRT (IF SET)
5513 020516 005737 002152 IFB233: TST RXXX ;IF DRIVE IS RXXX
5514 020522 001411 BEQ IFD233 ;THEN
5515 020524 032737 000002 002214 IFC233: BIT #2,UUT ;IF SIDE #1 IS SELECTED
5516 020532 001403 BEQ ELC233 ;THEN
5517 020534 052701 001000 BIS #1000,R1 ;SET SIDE #1 BIT
5518 020540 000402 BR IFD233 ;BRANCH TO IF 'D'
5519 020542 042701 001000 ELC233: BIC #1000,R1 ;SET FOR SIDE #0
5520 020546 032737 000001 002214 IFD233: BIT #1,UUT ;IF UNIT UNDER TEST IS
5521 020554 001403 BEQ ELD233 ;DRIVE #1
5522 020556 052701 000020 BIS #20,R1 ;THEN SET DRIVE #1 SELECT BIT
5523 020562 000402 BR EID233 ;BRANCH TO IF 'D'
5524 020564 042701 000020 ELD233: BIC #20,R1 ;ELSE CLEAR DRIVE #1 SELECT BIT
5525 020570 053701 002222 EID233: BIS DEN,R1 ;SET DENSITY BIT
5526 020574 052701 000100 BIS #100,R1 ;SET INTERRUPT BIT
5527 020600 010137 017526 MOV R1,DRVFN ;PASS UP FUNCTION WORD
5528 020604 000240 NOP ;
5529 020606 000207 END233: RTS PC ;RETURN
5530 ;-----
5531 020610 000000 DVTST: 0 ;DRIVE TEST WORD
5532 ;MOD 2.3.3 ----- END MODULE -----

```

.SBTTL MOD 2.3.4 - OUTPUT DRIVE FUNCTION

```

5534
5535
5536
5537 020612 000240          OTDVFN: NOP
5538 020614 013701 021252  MOV      CSADR,R1      ;GET STATUS REG ADR
5539 020620 062701 000002  ADD      #2,R1        ;ADD 2 TO ADR
5540 020624 010137 021250  MOV      R1,DBADR     ;SAVE AS DATA ADDRESS
5541 020630 012737 000040 021550  MOV      @DNBIT,RDYWD ;READY TEST WD (PASS TO 2.3.4.1)
5542 020636 013737 017524 021240  MOV      WDCT,WRDS    ;WORD FOR OUTPUT (PASS TO 2.3.4.1)
5543 020644 013737 021252 021242  MOV      CSADR,ADRS   ;ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
5544 020652 004737 021260  JSR      PC,OUTSMD   ;OUTPUT FUNCTION WD (FW) DO 2.3.4.1)
5545 020656 000240          NOP
5546 020660 032737 040000 017524  IFA234: BIT      #40000,WDOT ;IF FUNCTION IS
5547 030666 001001          BNE      ELA234      ;NOT AN "INITIALIZE" (FW BIT#14=0)
5548 020670 000402          BR       THA234      ;THEN 'A'
5549 020672 000137 021234  ELA234: JMP      END234     ;ENDIF 'A' -DONE
5550 020676 032737 000010 017524  THA234: BIT      #10,WDOT ;THEN, IF FUNCTION IS
5551 020704 001043          BNE      ELB234      ;"READ, WRITE, FILL, EMPTY" (FW BIT #3=0)
5552 020706 032737 000004 017524  IFH234: BIT      #4,WDOT  ;AND THEN IF FUNCTION IS
5553 020714 001050          BNE      ELH234      ;"EMPTY, FILL" (FW BIT#2=0)
5554 020716 012737 000200 021550  MOV      @TRBIT,RDYWD ;THEN SET OUTPUT READY TEST WORD (PASS TO 2.3.4.1)
5555 020724 013737 017522 021240  MOV      WDCT,WRDS    ;AND SET WORD FOR OUTPUT (PASS TO 2.3.4.1)
5556 020732 013737 021250 021242  MOV      DBADR,ADRS   ;AND SET ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
5557 020740 004737 021260  JSR      PC,OUTSMD   ;OUTPUT BASE ADDRESS WORD DO 2.3.4.1
5558 020744 032737 000002 017524  IFK234: BIT      #2,WDOT  ;IF "FILL" (FW BIT#1=0)
5559 020752 001004          BNE      ELK234      ;THEN
5560 020754 012737 035562 021240  MOV      @DATPAT,WRDS ;SET DATA PATTERN ADR (PASS TO 2.3.4.1)
5561 020762 000403          BR       EIK234      ;BR TO END IF 'K'
5562 020764 012737 036162 021240  ELK234: MOV      @DATBUF,WRDS ;SET DATA BUFFER ADR (PASS TO 2.3.4.1)
5563 020772 012737 000200 021550  EIK234: MOV      @TRBIT,RDYWD ;SET OUTPUT READY TEST WORD (PASS TO 2.3.4.1)
5564 021000 013737 021250 021242  MOV      DBADR,ADRS   ;ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
5565 021006 004737 021260  JSR      PC,OUTSMD   ;OUTPUT WORD COUNT WORD DO 2.3.4.1
5566 021012 000445          BR       EIH234      ;BRANCH TO END IF 'H'
5567 021014 000240          NOP
5568 021016 032737 000004 017524  IFC234: BIT      #4,WDOT  ;IF FUNCTION WORD IS
5569 021024 001455          BEQ      ELC234      ;"WRITE D.D" OR "READ E.C" (FW BIT #2=1)
5570 021026 032737 000002 017524  IFD234: BIT      #2,WDOT  ;THEN, IF FUNCTION IS
5571 021034 001035          BNE      ELD234      ;"WRITE D.D", THEN (FW BIT#1=0)
5572 021036 012737 000200 021550  ELH234: MOV      @TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
5573 021044 013737 021256 021240  MOV      SECADR,WRDS  ;MOVE TRACK AND SECTOR ADDRESS
5574 021052 042737 177700 021240  BIC      #177700,WRDS ;FORMAT TO SECTOR ADDRESS
5575 021060 013737 021250 021242  MOV      DBADR,ADRS   ;ADDRESS OF OUTPUT
5576 021066 004737 021260  JSR      PC,OUTSMD   ;OUTPUT SECTOR ADDRESS
5577 021072 013737 021254 021240  MOV      TRKADR,WRDS  ;MOVE TRACK AND SECTOR ADDRESS
5578 021100 042737 177600 021240  BIC      #177600,WRDS ;FORMAT TRACK ADDRESS
5579 021106 012737 000200 021550  MOV      @TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
5580 021114 013737 021250 021242  MOV      DBADR,ADRS   ;ADDRESS OF OUTPUT
5581 021122 004737 021260  JSR      PC,OUTSMD   ;OUTPUT TRACK ADDRESS
5582 021126 000440          BR       EIB234      ;ENDIF H -DONE
5583 021130 012737 000200 021550  ELD234: MOV      @TRBIT,RDYWD ;SET READY WD TO TR MODE
5584 021136 012737 002312 021240  MOV      @XERUUT,WRDS ;EXT ERR. CODE TABLE ADD
5585 021144 013737 021250 021242  MOV      DBADR,ADRS   ;ADDRESS OF OUTPUT, RXDB

```



```

5587 021152 004737 021260          JSR    PC,OUTSWD      ;O/P BASE ADD FOR ERR. CODE
5588 021156 000424          BR     EIB234        ;DONE
5589 021160 000240          ELC234: NOP          ;
5590 021162 032737 000002 017524  IFE234: BIT    #2,WDOT ; IF FUNCTION IS
5591 021170 001404          BEQ    ELE234        ; "READ STATUS" (FW BIT#1=1)
5592 021172 012737 000001 021244  THE234: MOV    #1,ERSTAT ; THEN-SET ERR STATUS FLAG
5593 021200 000413          BR     EIB234        ;DONE
5594 021202 012737 000200 021550  ELE234: MOV    @TRBIT,RDYWD ; SET OUTPUT READY TEST WD
5595 021210 013737 021246 021240  MOV    VALWD,WRDS    ; VALIDATION WORD
5596 021216 013737 021250 021242  MOV    DBADR,ADRS    ; ADDRESS OF OUTPUT, RXDB
5597 021224 004737 021260          JSR    PC,OUTSWD    ; OUTPUT VALIDATION WORD
5598 021230 004737 021324          EIB234: CALL   WATCH    ; CALL MOD U. 2 -WATCH DOG
5599 021234 000240          END234: NOP         ;
5600 021236 000207          RTS     PC           ; RETURN TO MOD 2.3
5601                                     ;-----
5602 021240 000000          WRDS:  0            ; MODULE 2.3.4.1 OUTPUT WORD
5603 021242 000000          ADRS:  0            ; MODULE 2.3.4.1 OUTPUT ADDRESS
5604 021244 000000          ERSTAT: 0          ; MODULE 0.0 ERR STATUS READ FLAG
5605 021246 000111          VALWD: 111         ; EXTERNAL, VALIDATION WD (SET DENS-ASCII "I")
5606 021250 000000          DBADR: 0           ; RX DATA BUFFER ADDRESS
5607 021252 000000          CSADR: 0           ; RX CONT/STATUS ADDRESS
5608 021254 000000          TRKADR: 0          ; TRACK ADDRESS
5609 021256 000000          SECADR: 0          ; SECTOR ADDRESS
5610                                     ; MOD 2.3.4 ----- END MODULE -----
5611
5612
5613
5614
5615                                     .SBTTL  MOD      2.3.4.1 OUTPUT SINGLE WORD
5616                                     ;-----
5617
5618 021260 000240          OUTSWD: NOP          ;
5619 021262 013737 021252 021552  MOV    CSADR,CSRADR  ; SET C&S REG ADR
5620 021270 013737 021550 021550  MOV    RDYWD,RDYWD   ; OUTPUT READY WORD (PASS TO DELAY)
5621 021276 004737 021450          JSR    PC,DELAY     ; DELAY FOR READY DO DELAY
5622 021302 033777 021550 177742  BIT    RDYWD,@CSADR  ; IF READY,
5623 021310 001404          BEQ    ED2341        ; THEN
5624 021312 000240          NOP                 ;
5625 021314 013777 021240 177720  MOV    WRDS,@ADRS    ; MOV WORD TO ADDRESS
5626 021322 000207          ED2341: RTS     PC   ; RETURN TO MOD 2.3.4
5627                                     ; MOD 2.3.4.1 ----- END MODULE -----
    
```

```
5629 . SBTTL MOD U. 2. 3. 4 - WATCH DOG TIMER
5630 ;
5631 ;
5632 021324 005037 021446 WATCH: CLR DNFLAG ; CLEAR DONE FLAG
5633 021330 SETPRI #PRI00 ; SET PROCESSOR PRI=0 - ALLOW INTERRUPTS
5634 021336 013704 021442 MOV DX,R4 ; SET DELAY MULT
5635 021342 013703 021444 BAU234: MOV DLY,R3 ; SET DELAY
5636 021346 005737 021446 IBU234: TST DNFLAG ; IF INTERRUPTS DONE FLAG
5637 021352 001410 BEQ LBU234 ; IS SET, THEN
5638 021354 032777 000040 177670 ICU234: BIT #DNBIT, @CSADR ; IF DONT BIT
5639 021362 001023 BNE XU234 ; IS NOT SET, THEN
5640 021364 012737 010000 002234 MOV #BIT12, ERRYP ; SET INTERR, BUT NO DONE ERROR
5641 021372 000417 BR XU234 ; BR TO MOD 'EXIT'
5642 021374 005303 LBU234: DEC R3 ; DECREMENT DELAY COUNT
5643 021376 001363 UDU234: BNE IBU234 ; DO UNIT DELAY COUNT=0
5644 021400 005304 DEC R4 ; DECREMENT DELAY MULT
5645 021402 001357 UAU234: BNE BAU234 ; DO UNTIL DELAY MULT=0
5646 021404 032777 000040 177640 IEU234: BIT #DNBIT, @CSADR ; IF DONE BIT IS
5647 021412 001404 BEQ LEU234 ; SET, THEN
5648 021414 052737 020000 002234 BIS #BIT13, ERRYP ; SET DONE, BUT NO INTERRUPT ERROR
5649 021422 000403 BR XU234 ; BR TO MOD 'EXIT'
5650 021424 052737 040000 002232 LEU234: BIS #BIT14, SYSERR ; SET T. O. ERROR
5651 021432 XU234: SETPRI #PRI07 ; SET PROCESSOR PRI=7 - NO INTERRUPTS
5652 021440 000207 RTS PC ; RETURN TO MOD 2. 3. 4
```

```
5653 ;
5654 021442 000010 DX: 10 ; DELAY MULT
5655 021444 100000 DLY: 100000 ; DELAY
5656 021446 000000 DNFLAG: 0 ; DONE FLAG
5657 ;
5658 ;
5659 . SBTTL MOD U. 2. 3. 4 ---- END MODULE ----
5660 ;
```

```
5661 . SBTTL MOD U. 2. 3/4 DELAY
5662 ;
5663 021450 000240 DELAY: NOP ;
5664 021452 023727 021550 000000 IFAU23: CMP RDYWD, #0 ; IF READY WORD
5665 021460 001430 BEQ XU23 ; EQUALS ZERO, THEN BR TO END IF 'A'
5666 021462 013704 021544 MOV RYDX, R4 ; SET READY DELAY MULT
5667 021466 013703 021546 BDAU23: MOV RYDLY, R3 ; SET READY DELAY
5668 021472 033777 021550 000052 BDBU23: BIT RDYWD, @CSRADR ; IF READY
5669 021500 001020 BNE XU23 ; EQUAL TO "1", THEN BR TO END IF 'B'
5670 021502 005303 DEC R3 ; ELSE DECREMENT DELAY
5671 021504 001372 BNE BDBU23 ; DO UNTIL R3=0
5672 021506 005304 DEC R4 ; DECREMENT DELAY MULT.
5673 021510 001366 BNE BDAU23 ; DO UNTIL R4=0
5674 021512 052737 040000 002232 BIS #40000, SYSERR ; SET TIME OUT ERR
5675 021520 017737 000026 002236 MOV @CSRADR, CSRUUT ; GET UUT C&S REG
5676 021526 062737 000002 021552 ADD #2, CSRADR ; SET CSRADR TO DB REG
5677 021534 017737 000012 002240 MOV @CSRADR, ESRUUT ; GET UUT E&S REG
5678 XU23: RTS PC ; RETURN TO CALLING MOD
```

```
5679 ;
5680 021544 000020 RYDX: 20 ; READY MULTIPLIER
5681 021546 100000 RYDLY: 100000 ; READY DELAY
5682 021550 000000 RDYWD: 0 ; READY WORD - TEST FOR DEVICE READY
5683 021552 000000 CSRADR: 0 ; C&S REG OF UNIT- WAITING FOR
; MOD U. 2. 3. 4 ---- END MODULE ----
```

```
5685  
5686 . SBTTL MOD 2.4 - EVALUATE TEST RESULTS  
5687 ;  
5688 -----  
5689 021554 013737 021634 023414 EVTSTR: MOV TSTEV,FUNEV ;PASS TEST FUNCTION  
5690 021562 000240 NOP ;  
5691 021564 004737 022432 CALL EVDVST ;CALL MOD 2.4.2 - EVALUATE DRIVE STATE  
5692 021570 013737 021634 025026 MOV TSTEV,FNEV4 ;PASS TEST FUNCTION  
5693 021576 004737 024702 CALL EVUTEC ;CALL MOD 2.4.4 - EVAL UNIT ERR CODE  
5694 021602 032737 020000 021634 IFA24: BIT #20000,TSTEV ;IF DATA CK BIT  
5695 021610 001402 BEQ EIA24 ;IS SET, THEN  
5696 021612 004737 021636 CALL EVDATA ;CALL MOD 2.4.1 - EVALUATE DATA  
5697 021616 013737 021634 024146 EIA24: MOV TSTEV,TSTCK ;PASS DRIVE TEST  
5698 021624 004737 023556 CALL UPDVST ;CALL MOD 2.4.3 UPDATE DRIVE STATISTICS  
5699 021630 000240 NOP ;  
5700 021632 000207 RTS PC ;  
5701 ;  
5702 021634 000000 TSTEV: 0  
5703 ;MOD 2.4 ----- END MODULE -----  
5704 ;
```

```

SBTTL MOD 2.4.1 - EVALUATE DATA
-----
5706
5707
5708
5709 021636 005037 022300      EVDATA: CLR      DAERCT      ;CLEAR DATA ERR COUNT
5710 021642 005037 022272      CLR      SEEKCK      ;CLEAR SEEK CK
5711 021646 012737 000001 022306  MOV      #1,PTHEAD    ;SET PRINT HEADER FLAG
5712 021654 013701 002270      MOV      WDCNT,R1     ;SAVE WORD COUNT
5713 021660 006301      ASL      R1           ;
5714 021662 162701 000001      SUB      #1,R1        ;SUBTRACT 2 TO GET CHECKSUM
5715 021666 012702 035562      MOV      #DATPAT,R2   ;GET ADDRESS DATA SOURCE
5716 021672 012703 036162      MOV      #DATBUF,R3   ;GET ADDRESS DATA BUFFER
5717 021676 060102      ADD      R1,R2        ;CAL. ADDR SOURCE CHECKSUM
5718 021700 060103      ADD      R1,R3        ;CAL. ADDR BUFFER CHECKSUM
5719 021702 121213      IFA241: CMPB      (R2),(R3) ;IF CHECK SUMS
5720 021704 001407      BEQ      ELA241      ;NOT= THEN
5721 021706 032737 000002 015670  IFI241: BIT      #2,ERTSAV ;IF CRC ERR
5722 021714 001003      BNE      ELA241      ;NOT SET, THEN
5723 021716 052737 000004 002234  BIS      #4,ERRTYP    ;SET CHECKSUM ERR
5724 021724 005037 022276      ELA241: CLR      BYTNUM ;CLEAR BYTE NUMBER
5725 021730 162701 000001      SUB      #1,R1        ;CAL. TOTAL BYTE COUNT-LAST TWO
5726 021734 010137 022274      MOV      R1,BYTCNT    ;SAVE BYTE COUNT
5727 021740 012701 035562      BDA241: MOV      #DATPAT,R1 ;SET TEMP#1=DATA SOURCE BEGIN ADR
5728 021744 012702 036162      MOV      #DATBUF,R2  ;SET TEMP#2=DATA BUFFER BEGIN ADR
5729 021750 063701 022276      ADD      BYTNUM,R1    ;CAL CURRENT BYTE ADDR (SOURCE)
5730 021754 063702 022276      ADD      BYTNUM,R2    ;CAL CURRENT BYTE ADDR (BUFFER)
5731 021760 121112      CMPB      (R1),(R2)  ;IF SOURCE BYTE & BUFFER BYTE
5732 021762 001502      BEQ      ELB241      ;NOT EQUAL
5733 021764 005237 022300      INC      DAERCT      ;INCREMENT DATA ERR COUNT
5734 021770 052737 000010 002234  BIS      #10,ERRTYP  ;SET DATA ERR-ERR TYPE
5735 021776 042737 000004 002234  BIC      #4,ERRTYP    ;CLR CK SUM ERR-ERR TYPE
5736 022004 023727 022276 000002  IFC241: CMP      BYTNUM,#2 ;IF BYTE #0 OR #1
5737 022012 002006      BGE      IFE241      ;THEN
5738 022014 005737 022276      IFD241: TST      BYTNUM ;IF BYTE #0
5739 022020 001003      BNE      IFE241      ;THEN
5740 022022 052737 000001 022272  BIS      #1,SEEKCK    ;SET SEEK ERR-ERR TYPE
5741 022030 023727 022300 000012  IFE241: CMP      DAERCT,#12 ;IF OVER 10 DATA ERRORS
5742 022036 103404      BLO      THF241      ;THEN
5743 022040 032737 000020 002164  IFF241: BIT      #20,SWREG ;IF PRINT ONLY 10 DATA ERROR FLAG
5744 022046 001047      BNE      EIF241      ;IS NOT SET, THEN
5745 022050 111137 022302      THF241: MOVB     (R1),DATASB
5746 022054 111237 022304      MOVB     (R2),DATAWS
5747 022060 005737 022306      IFM241: TST      PTHEAD ;IF PRINT HEADER
5748 022064 001420      BEQ      EIM241      ;OK, THEN
5749 022066 005037 022306      CLR      PTHEAD      ;CLEAR PRINT HEADER
5750 022072      PRINTB #MSG1,UNIT,TRACK,SECTOR
5751 022126      EIM241: PRINTB #MSG2,BYTNUM,<B,DATASB>,<B,DATAWS>
5752 022166 000240      EIF241: NOP          ;
5753 022170 005237 022276      ELB241: INC      BYTNUM ;INCREMENT BYTE #
5754 022174 005337 022274      DEC      BYTCNT      ;DECREMENT BYTE COUNT
5755 022200 005737 022274      TST      BYTCNT      ;DO UNTIL BYTE COUNT
5756 022204 003255      BGT      BDA241      ;EQUALS 0
5757 022206 005737 022272      IFJ241: TST      SEEKCK ;IF DISK SEEK ERR
5758 022212 001413      BEQ      END241      ;IS SET AND
5759 022214 032737 000010 002234  IFK241: BIT      #10,ERRTYP ;IF DATA ERR
5760 022222 001007      BNE      END241      ;NOT SET AND
5761 022224 032737 000002 015670  IFL241: BIT      #2,ERTSAV ;IF CRC ERR
    
```



```

5788          . SBTTL MOD 2.4.2 - EVALUATE DRIVE STATE
5789          ;-----
5790
5791 022432 013705 002216          EVDVST: MOV      UUTADR,R5
5792 022436 013737 002236 023416  MOV      CSRUUT,CSREV ;GET COMMAND & STATUS LAST OP UUT
5793 022444 013737 002240 023420  MOV      ESRUUT,ESREV ;GET ERROR STATUS LAST OP UUT
5794 022452 005037 002312          CLR      XERUUT ;CLEAR EXTENDED ERROR CODE LOCATION
5795 022456 032737 000040 023416  IFA242: BIT      #40,CSREV ;IF DONE NOT
5796 022464 001033          BNE     IFB242 ;SET THEN
5797 022466 012715 040000          MOV     #40000,(R5) ;ISSUE PROG INIT TO UUT
5798 022472 000240          NOP
5799 022474 013737 002216 021552  MOV     UUTADR,CSRADR ;SET CSR ADR
5800 022502 012737 000040 021550  MOV     #DNBIT,RDYWD ;SET DONE TEST
5801 022510 004737 021450          CALL   DELAY ;WAIT FOR TR
5802 022514 032715 000040          IFC242: BIT      #40,(R5) ;IF DONE NOT
5803 022520 001005          BNE     ELC242 ;SET THEN
5804 022522 052737 000010 002232  BIS     #10,SYSEERR ;SET NO DONE ON INT-SYS ERR
5805 022530 000137 023406          JMP     END242 ;BR TO END MOD
5806 022534 113701 023414          ELC242: MOVB    FUNEV,R1 ;GET DRIVE FUNCTION
5807 022540 042701 177770          BIC    #177770,R1 ;CLEAR ALL BUT FUNCTION
5808 022544 050137 002232          BIS     R1,SYSEERR ;SET NO DONE ON FUNCTION-SYS ERR
5809 022550 000137 023406          JMP     END242 ;BR TO END MOD
5810 022554 004737 023422          IFB242: CALL   EVDVRE ;CALL MOD 2.4.2.1 EVALUATE DRIVE RESPONSE
5811 022560 005737 002232          TST    SYSEERR ;IF SYS ERR
5812 022564 001463          BEQ    IFG242 ;NOT=0 THEN
5813 022566 032737 000001 002214  BIT     #1,UUT ;IFDRV#1 UNDER TST
5814 022574 001404          BEQ    15 ;THEN
5815 022576 012737 000020 023412  MOV     #20,EVCMD ;SET CMD TO DRV#1
5816 022604 000402          BR     25 ;BR
5817 022606 005037 023412          15:    CLR     EVCMD ;SET CMD TO DRV#0
5818 022612 052737 000013 023412  25:    BIS     #13,EVCMD ;SET READ UUT ESR IN CMD
5819 022620 053737 002222 023412  BIS     DEN,EVCMD ;SET DEN FOR CMD
5820 022626 013715 023412          MOV     EVCMD,(R5) ;READ UUT ESR
5821 022632 013737 002216 021552  MOV     UUTADR,CSRADR ;SET CSR ADR
5822 022640 012737 000040 021550  MOV     #DNBIT,RDYWD ;SET DONE BIT
5823 022646 004737 021450          CALL   DELAY ;CALL
5824 022652 032715 000040          IFX242: BIT      #40,(R5) ;IF DONE BIT
5825 022656 001005          BNE     IFD242 ;NOT SET THEN
5826 022660 052737 000200 002232  BIS     #200,SYSEERR ;SET NO DONE BIT (SECONDARY PROBLEM)
5827 022666 000137 023406          JMP     END242 ;BK TO END
5828 022672 032715 100000          IFD242: BIT      #100000,(R5) ;IF ERR BIT
5829 022676 001403          BEQ    IFE242 ;SET
5830 022700 052737 100000 002234  BIS     #100000,ERRTYP ;ERR BIT - ERR TYPE
5831 022706 013701 002216          IFE242: MOV     UUTADR,R1 ;GET UUT ADR
5832 022712 062701 000002          ADD    #2,R1 ;CAL DBR ADR
5833 022716 032711 000200          BIT     #200,(R1) ;IF DRV RDY BIT
5834 022722 001102          BNE     IFN242 ;EQUALS 0
5835 022724 052737 000040 002232  BIS     #40,SYSEERR ;SET DRIVE NOT RDY-SYS ERR
5836 022732 000561          BR     IFS242 ;BR TO END IF 'E'
5837 022734 032737 002021 023420  IFG242: BIT      #2021,ESREV ;IF ANY ESR ERR BIT SET
5838 022742 001410          BEQ    IFH242 ;THEN
5839 022744 032737 100000 023416  IFI242: BIT      #100000,CSREV ;IF UUT ERR BIT
5840 022752 001010          BNE     IFJ242 ;NOT=1 THEN
5841 022754 052737 040000 002234  BIS     #40000,ERRTYP ;SET MISSING ERR BIT
5842 022762 000450          BR     IFL242 ;BR TO IF 'L'
5843 022764 032737 100000 023416  IFH242: BIT      #100000,CSREV ;IF UUT CSR ERR BIT

```

5844	022772	001456				BEQ	IFN242		;EQUALS 1 THEN
5845	022774	013701	021634		IFJ242:	MOV	TSTEV,R1		;GET TEST FUNCTION
5846	023000	042701	177774			BIC	#177774,R1		;CLEAR ALL BUT TWO BOTTOM BITS
5847	023004	022701	000002			CMP	#2,R1		;IF WRITE FUNCTION
5848	023010	001004				BNE	IFK242		;THEN
5849	023012	052737	004000	002234		BIS	#4000,ERRTYP		;SET WRITE ERR - ERR TYPE
5850	023020	000431				BR	IFL242		;BR TO IF 'L'
5851	023022	013701	021634		IFK242:	MOV	TSTEV,R1		;GET TEST FUNCTION
5852	023026	042701	177770			BIC	#177770,R1		;CLEAR ALL BUT FUNCTION
5853	023032	022701	000003			CMP	#3,R1		;IF READ FUNCTION
5854	023036	001004				BNE	IFM242		;THEN
5855	023040	052737	002000	002234		BIS	#2000,ERRTYP		;SET READ ERR-ERR TYPE
5856	023046	000416				BR	IFL242		;BR TO IF 'L'
5857	023050	013701	021634		IFM242:	MOV	TSTEV,R1		;GET TEST FUNCTION
5858	023054	042701	177771			BIC	#177771,R1		;CLEAR BITS
5859	023060	032701	000006			BIT	#6,R1		;IF FILL/EMPTY BUFFER
5860	023064	001004				BNE	ELM242		;THEN
5861	023066	052737	001000	002234		BIS	#1000,ERRTYP		;SET FILL/EMPTY ERR-ERR TYP
5862	023074	000403				BR	IFL242		;BR TO IF 'L'
5863	023076	052737	000400	002234	ELM242:	BIS	#400,ERRTYP		;ELSE SET UNK ERR
5864	023104	032737	000001	023420	IFL242:	BIT	#1,ESREV		;IF CRC ERR (ESR)
5865	023112	001406				BEQ	IFN242		;THEN
5866	023114	042737	000001	002234		BIC	#1,ERRTYP		;CLEAR ANY SEEK ERR
5867	023122	052737	000002	002234		BIS	#2,ERRTYP		;SET CRC ERR
5868	023130	032737	006010	023420	IFN242:	BIT	#6010,ESREV		;IF ESR BIT#3,10,11 ARE
5869	023136	001404				BEQ	IFF242		;SET, THEN
5870	023140	052737	100000	002232		BIS	#100000,SYSEERR		;SET UNKNOWN ERR-SYS ERR
5871	023146	000453				BR	IFS242		;BR TO IF 'S'
5872	023150	013701	021634		IFF242:	MOV	TSTEV,R1		;GET TEST FUNCTION
5873	023154	032701	000002			BIT	#2,R1		;IF FUNCTION WAS
5874	023160	001425				BEQ	IFR242		;POSSIBLE READ OR WRITE
5875	023162	032701	000004			BIT	#4,R1		;BUT REALLY
5876	023166	001022				BNE	IFR242		;IS READ OR WRITE, THEN
5877	023170	005737	002224		IF0242:	TST	DELDAT		;IF DELETED DATA FLAG IS
5878	023174	001410				BEQ	IFQ242		;SET THEN
5879	023176	032737	000100	023420	IFP242:	BIT	#100,ESREV		;IF UUT ESR DD BIT
5880	023204	001013				BNE	IFR242		;NOT SET THEN
5881	023206	052737	000100	002234		BIS	#100,ERRTYP		;SET MISSING DP MARK-ERR TYP
5882	023214	000407				BR	IFR242		;BR TO IF 'R'
5883	023216	032737	000100	023420	IFQ242:	BIT	#100,ESREV		;IF D. D. BIT IS
5884	023224	001403				BEQ	IFR242		;SET THEN
5885	023226	052737	000040	002234		BIS	#40,ERRTYP		;SET UNEX DD BIT
5886	023234	032737	000020	023420	IFR242:	BIT	#20,ESREV		;IF DEN. ERR. (ESR)
5887	023242	001403				BEQ	IFU242		;THEN
5888	023244	052737	020000	002232		BIS	#20000,SYSEERR		;SET DEN. ERR-SYS ERR
5889	023252	005737	002222		IFU242:	TST	DEN		;IF DOUBLE DEN MODE IS
5890	023256	001407				BEQ	IFS242		;SET AND THEN
5891	023260	032737	000040	023420	IFV242:	BIT	#40,ESREV		;IF UUT RESPONDS IN
5892	023266	001003				BNE	IFS242		;SINGLE DENSITY, THEN
5893	023270	052737	010000	002232		BIS	#10000,SYSEERR		;SET DRIVE DENSITY ERR-SYS ERR
5894	023276	032737	100000	023416	IFS242:	BIT	#100000,CSREV		;IF UUT ERR BIT
5895	023304	001440				BEQ	END242		;NOT=0 THEN
5896	023306	012737	000001	002310		MOV	#1,PRTECD		;SET PRINT ERROR CODE FLAG
5897	023314	012737	000017	023412	THS242:	MOV	#17,EVCMD		;SET UUT EXTENDED ERROR CODE
5898	023322	053737	002222	023412		BIS	DEN,EVCMD		;SET DEN FOR CMD
5899	023330	013715	023412			MOV	EVCMD,(R5)		;GET UUT EXT. ERROR CODE REGS

```
5900 023334 013701 002216      MOV      UUTADR,R1      ;GET UUT ADDR
5901 023340 062701 000002      ADD      #2,R1        ;CAL DATA ADR
5902 023344 013737 002216 021552  MOV      UUTADR,CSRADR ;SET CSR ADR
5903 023352 012737 000200 021550  MOV      #TRBIT,RDYWD  ;SET "TR" BIT TEST
5904 023360 004737 021450      CALL     DELAY         ;CALL DELAY MODULE-WAIT FOR TR
5905 023364 032715 000200      IFT242: BIT      #200,(R5) ;IF TR
5906 023370 001004      BNE     ELT242        ;NOT SET
5907 023372 052737 040007 002232  BIS     #40007,SYSERR ;THEN SET "TR" ERR ON FUNCTION
5908 023400 000402      BR      END242        ;BR TO END MOD
5909 023402 012711 002312  ELT242: MOV      #XERUUT,(R1) ;SEND BASE ADDR FOR EXTEND ERR CODE
5910 023406 000240      END242: NOP
5911 023410 000207      RTS      PC
5912
5913 023412 000000      EVCMD:  0              ;CMD WORD USED IN THIS MOD
5914 023414 000000      FUNEV:  0
5915 023416 000000      CSREV:  0
5916 023420 000000      ESREV:  0
5917
5918      ;MOD 2.4.2 ----- END MODULE -----
```


5920
 5921
 5922
 5923
 5924
 5925
 5926
 5927
 5928
 5929
 5930
 5931
 5932
 5933
 5934
 5935
 5936
 5937
 5938
 5939
 5940
 5941
 5942
 5943
 5944
 5945
 5946
 5947
 5948
 5949
 5950
 5951
 5952

023422 013701 021634
 023426 042701 177771
 023432 032701 000006
 023436 001445
 023440 005737 002152
 023444 001421
 023446 032737 000002 002214
 023454 001403
 023456 012701 001000
 023462 000401
 023464 005001
 023466 013702 002240
 023472 042702 176777
 023476 020102
 023500 001403
 023502 052737 001000 002232
 023510 032737 000001 002214
 023516 001403
 023520 012701 000400
 023524 000401
 023526 005001
 023530 013702 002240
 023534 042702 177377
 023540 020102
 023542 001403
 023544 052737 000400 002232
 023552 000240
 023554 000207

```

.SBTTL MOD 2.4.2.1 - EVALUATE DRIVE RESPONSE
-----
EVDVRE: MOV     TSTEV,R1      ;GET TEST FUNCTION
          BIC     #177771,R1   ;CLEAR BITS
          BIT     #6,R1        ;IF NOT FILL/EMPTY BUFFER
          BEQ     6$           ;THEN
          TST     RXXX         ;IF RXXX
          BEQ     1$           ;AND
          BIT     #2,UUT       ;SIDE # SELECTED
          BEQ     2$           ;THEN
          MOV     #1000,R1     ;SET R1 TO TEST SIDE #1 SELECT
          BR      3$           ;BR TO TEST RESPONSE
2$:      CLR     R1            ;SET R1 TO TEST SIDE #0 SELECT
3$:      MOV     ESRUUT,R2     ;GET ESR UNIT UNDER TEST
          BIC     #176777,R2   ;CLEAR ALL BITS BUT SIDE SELECT
          CMP     R1,R2        ;IF SIDE SELECT
          BEQ     1$           ;NOT=SIDE RESPONDING THEN
          BIS     #1000,SYSErr ;SET WRONG SIDE RESPONDING SYS ERR
1$:      BIT     #1,UUT       ;IF DRIVE #1 SELECTED
          BEQ     4$           ;THEN
          MOV     #400,R1      ;SET R1 TO TEST DRIVE #1 SEL
          BR      5$           ;BR TO TEST RESPONSE
4$:      CLR     R1            ;SET R1 TO TEST DRIVE #0 SEL
5$:      MOV     ESRUUT,R2     ;GET ESR UNIT UNDER TEST
          BIC     #177377,R2   ;CLEAR ALL BITS BUT DRIVE RESPONDING
          CMP     R1,R2        ;
          BEQ     6$           ;
          BIS     #400,SYSErr  ;SET WRONG DRIVE RESPONDING SYS ERR
6$:      NOP
          RTS     PC
;MOD 2.4.2.1 ----- END MODULE -----
  
```

```

5954 . SBTTL MOD 2.4.3 - UPDATE DRIVE STATISTICS
5955 ;-----
5956
5957 023556 000240 UPDVST: NOP ;
5958 023560 032737 000002 024126 IA243: BIT #2,ETSAV ; IF ERR TYP SAVE
5959 023566 001405 BEQ EA243 ; HAS CRC ERR BIT SET, THEN
5960 023570 004737 024254 CALL UDCRST ; CALL UPDATE CRC STATISTICS
5961 023574 005037 024126 CLR ETSAV ; CLEAR ERR TYPE SAVE
5962 023600 000457 BR IG243 ; BR TO IF 'G'
5963 023602 013737 002234 024126 EA243: MOV ERR TYP,ETSAV ; SAVE ERR TYP --> ETSAV
5964 023610 013737 002234 024134 MOV ERR TYP,STERRG ; GET ERR TYP --> STAT ERR REG
5965 023616 005037 024136 CLR STCNTR ; ZERO STAT COUNTER
5966 023622 032737 000002 024134 ID243: BIT #2,STERRG ; IF ERR IS
5967 023630 001403 BEQ BF243 ; CRC, THEN
5968 023632 042737 006002 024134 BIC #6002,STERRG ; CLEAR CRC, RD, & WRT ERR BITS OF STAT ERR REG
5969 023640 000241 BF243: CLC ; CLEAR CARRY BIT
5970 023642 006037 024134 ROR STERRG ; ROTATE RIGHT STAT ERROR REG
5971 023646 103026 IB243: BCC EB243 ; IF CARRY BIT SET, THEN
5972 023650 013701 024136 MOV STCNTR,R1 ; GET STAT COUNTER
5973 023654 006301 ASL R1 ; & DOUBLE FOR WORD ADDRESSING
5974 023656 062701 024150 ADD #ETTAB,R1 ; CAL. CLASSIFICATION WORD-ADDRESS
5975 023662 011137 024140 MOV (R1),CLASWD ; GET CLASSIFICATION WORD
5976 023666 011102 MOV (R1),R2 ; GET CLASSIFICATION WORD-TO FIND LOG OFFSET
5977 023670 000302 SHAB R2 ; GET CLASSIFICATION WORD UPPER BYTE
5978 023672 006302 ASL R2 ; --SHIFT LEFT TO GET LOG REG OFFSET (LAST 6 BITS)
5979 023674 006302 ASL R2 ; --SHIFT LEFT AGAIN
5980 023676 042702 177004 BIC #177004,R2 ; CLEAR UNWANTED BITS
5981 023702 010237 024142 MOV R2,LOGOFF ; SAVE ERROR LOG OFFSET
5982 023706 005711 IC243: TST (R1) ; IF ERR TYP CLASSIFICATION WORD
5983 023710 100403 BMI LC243 ; TYPE=SOFT, THEN
5984 023712 004737 024426 CALL UDSFST ; CALL UPDATE SOFT ERROR STATISTICS
5985 023716 000402 BR EB243 ; BR TO END 'B'
5986 023720 004737 024210 LC243: CALL UDHDST ; CALL UPDATE HARD ERROR STATISTICS
5987 023724 005237 024136 EB243: INC STCNTR ; INCREMENT STAT COUNTER
5988 023730 022737 000020 024136 UF243: CMP #16,STCNTR ; DO UNTIL ALL 16
5989 023736 101340 BHI BF243 ; BITS ARE DONE
5990 023740 013703 002312 IG243: MOV XERUT,R3 ; GET EXTENDED ERROR CODE
5991 023744 042703 177400 BIC #177400,R3 ; CLEAR UPPER BYTE
5992 023750 005703 TST R3 ; IF EXTENDED ERROR CODE
5993 023752 001410 BEQ IH243 ; NOT=0, THEN
5994 023754 162703 000010 SUB #10,R3 ; ADJ ERROR CODE # FOR LOGGING
5995 023760 012702 004672 MOV #ECLOG,R2 ; GET LOC OF ERR CODE LOG
5996 023764 060302 ADD R3,R2 ; ADD ERR CODE TO LOC ERR CODE LOG
5997 023766 063702 002220 ADD UUTOFF,R2 ; FIND LOC ERR REG THIS UNIT
5998 023772 005212 INC (R2) ; INCREMENT UNIT ERR REG
5999 023774 013703 002234 IH243: MOV ERR TYP,R3 ; GET ERR TYPE
6000 024000 042703 171774 BIC #171774,R3 ; CLEAR ALL ERRS BUT RD, WT, CRC, SEEK
6001 024004 005703 TST R3 ; IF ONE OF THESE ERRORS
6002 024006 001412 BEQ I1243 ; THEN
6003 024010 013702 002272 MOV TRACK,R2 ; GET TRACK ADR
6004 024014 006302 ASL R2 ; DOUBLE TRACK ADR FOR WORD ADDRESSING
6005 024016 006302 ASL R2 ; ADJ TRK
6006 024020 006302 ASL R2 ; FOR ADR.
6007 024022 062702 005156 ADD #TKXX,R2 ; ADD TRACK LOG LOCATION
6008 024026 063702 002220 ADD UUTOFF,R2 ; FIND LOC ERR REG THIS UNIT
6009 024032 005212 INC (R2) ; INCREMENT UNIT ERR REG
  
```

```

6010 024034 005737 024130      11243: TST      ERRSAV      ; IF ERR SAVE HAS
6011 024040 001023              BNE      L1243      ; NO ERROR SET, THEN
6012 024042 005237 024132      INC      ERSVCT      ; INCREMENT ERROR SAVE COUNTER
6013 024046 022737 000004 024132 1J243: CMP      #4,ERSVCT  ; IF ERROR SAVE COUNTER
6014 024054 101017              BHI      E1243      ; NOT=4, THEN
6015 024056 012701 002244      MOV      #SEEKRT,R1 ; SET BEGIN ADDRESS OF RETRY COUNTERS
6016 024062 012702 000011      MOV      #11,R2     ; SET # OF RETRY COUNTERS
6017 024066 005021      BK243: CLR      (R1)+  ; CLEAR RETRY COUNTER
6018 024070 005302              DEC      R2         ; DECREMENT RETRY COUNTER #
6019 024072 005702      UK243: TST      R2         ; DO UNTIL
6020 024074 001374              BNE      BK243      ; ALL COUNTERS CLEARED
6021 024076 005037 024132      CLR      ERSVCT      ; CLEAR ERROR SAVE COUNTER
6022 024102 005037 002242      CLR      RETRY       ; CLEAR RETRY COUNTER
6023 024106 000402              BR       E1243      ; BR TO END 'I'
6024 024110 005037 024132  L1243: CLR      ERSVCT      ; CLEAR ERROR SAVE COUNT
6025 024114 013737 002234 024130  E1243: MOV      ERRTP,ERRSAV ; SAVE ERROR TYPE FOR NEXT ERROR CHECK
6026 024122 000240      END243: NOP              ;
6027 024124 000207              RTS      PC         ; RETURN
6028
6029 024126 000000      ETSAV: 0              ; ERR TYPE SAVE
6030 024130 000000      ERRSAV: 0            ; ERR TYPE SAVE REG
6031 024132 000000      ERSVCT: 0            ; ERROR SAVE COUNTER-COUNTS # OF NO ERROR PASSES
6032 024134 000000      STERRG: 0           ; STAT ERR REG
6033 024136 000000      STCNTR: 0           ; STAT COUNTER
6034 024140 000000      CLASWD: 0           ; ERROR CLASSIFICATION WORD-FROM TABLE
6035 024142 000000      LOGOFF: 0           ; ERROR LOG OFFSET FROM #CKSML
6036 024144 000000      RTOFF: 0            ; RETRY COUNTER OFFSET FROM #SEEKRT
6037 024146 000000      TSTCK: 0            ; TEST WORD-USED TO CHECK TEST DONE
6038
; MOD 2.4.3 ----- END MODULE -----

```

```

6039
6040
6041
6042
6043 024150 005001      ETTAB: . WORD 005001  ; SFT /SEEK /SEEK /SK-RTMSK/ 0
6044 024152 006005      . WORD 006005  ; SFT /CRC /CRC /CRC / 1
6045 024154 100407      . WORD 100407  ; HRD /CKSML / - /HD / 2
6046 024156 012106      . WORD 012106  ; SFT /DATA /DATA /DT-RTMSK/ 3
6047 024160 154400      . WORD 154400  ; HRD / - / - / - / 4
6048 024162 113227      . WORD 113227  ; HRD /DDUNX /DD /HD / 5
6049 024164 113227      . WORD 113227  ; HRD /DOMIS /DD /HD / 6
6050 024166 154400      . WORD 154400  ; HRD / - / - / - / 7
6051 024170 154400      . WORD 154400  ; HRD /UNK / - / - / 8
6052 024172 101407      . WORD 101407  ; HRD /FIL-EMP/ - /HD / 9
6053 024174 010164      . WORD 010164  ; SFT /RD /RD /RD-RTMSK/ 10
6054 024176 011202      . WORD 011202  ; SFT /WRT /WT /WT-RTMSK/ 11
6055 024200 103407      . WORD 103407  ; HRD /INTR-ND/ - /HD / 12
6056 024202 104407      . WORD 104407  ; HRD /D-NINTR/ - /HD / 13
6057 024204 102407      . WORD 102407  ; HRD /ER-NSET/ - /HD / 14
6058 024206 154407      . WORD 154407  ; HRD /ERR BIT/ - /HD / 15
6059
6060
6061
6062
6063
6064
;-----<CLASSIFICATION (SEEK=1/CRC=5/DATA=6/WRITE=2/READ=4)
;-----<RETRY COUNTER OFFSET
;-----<LOG REGISTER OFFSET-(FROM CKSML ADDRESS)
;-----<TYPE (SOFT=0/HARD=1)
;-----

```

```

6066      .SBTTL MOD 2.4.3.1 - UPDATE HARD ERROR STATISTICS
6067      ;-----
6068
6069      024210 000240 UDHDST: NOP ;
6070      024212 032737 000007 024140 IA2431: BIT #7,CLASWD ; IF ERROR CLASS WORD-
6071      024220 001011 BNE LA2431 ; CLASS=HD(7), THEN
6072      024222 013701 024142 MOV LOGOFF,R1 ; GET ERROR LOG OFFSET
6073      024226 062701 004442 ADD #CKSML,R1 ; ERR LOG ADR=ERR LOG OFF + CKSML ADR
6074      024232 063701 002220 ADD UUTOFF,R1 ; UUT ERR LOG ADR=UUT OFFSET + ERR LOG ADR
6075      024236 005211 INC (R1) ; INCREMENT THE ERROR LOG
6076      024240 000240 NOP ;
6077      024242 000402 BR EA2431 ; BR TO END 'A'
6078      024244 000240 LA2431: NOP ;
6079      024246 000240 NOP ;
6080      024250 000240 EA2431: NOP ;
6081      024252 000207 X2431: RTS PC ; RETURN
6082      ;MOD 2.4.3.1 ----- END MODULE -----
6083
6084
6085
6086      .SBTTL MOD 2.4.3.2 - UPDATE CRC STATISTICS
6087      ;-----
6088
6089      024254 000240 UDCRST: NOP ;
6090      024256 032737 020000 024146 IA2432: BIT #BIT13,TSTCK ; IF TEST=DATA CHECK
6091      024264 001425 BEQ LA2432 ; BIT SET, THEN
6092      024266 032737 000010 002234 IB2432: BIT #BIT03,ERRTYP ; IF ERR TYPE=DATA ERR
6093      024274 001007 BNE LB2432 ; NOT SET, THEN
6094      024276 012737 000020 024142 MOV #20,LOGOFF ; SET LOG OFFSET=CRC BAD LOG
6095      024304 012737 000006 024144 MOV #6,RTOFF ; SET RETRY OFFSET=CRC ERR
6096      024312 000420 BR IC2432 ; BR TO 'C'
6097      024314 012737 000050 024142 LB2432: MOV #50,LOGOFF ; SET DATA LOG OFFSET
6098      024322 005037 024700 CLR RTMASK ; CLEAR RETRY MASK
6099      024326 012737 000012 024144 MOV #12,RTOFF ; SET DUMMY DATA RETRY COUNTER OFFSET
6100      024334 004737 024556 CALL SFERLG ; CALL SOFT ERROR LOGGER
6101      024340 012737 000010 024142 LA2432: MOV #10,LOGOFF ; SET LOG OFFSET=CRC ERR LOG
6102      024346 012737 000006 024144 MOV #6,RTOFF ; SET RETRY OFFSET=CRC ERR
6103      024354 032737 010000 024146 IC2432: BIT #BIT12,TSTCK ; IF READ AFTER WRITE (RAW)
6104      024362 001407 BEQ LC2432 ; BIT SET, THEN
6105      024364 012737 000020 024700 MOV #BIT04,RTMASK ; SET RETRY MASK=CRC
6106      024372 052737 000002 024700 BIS #BIT1,RTMASK ; SET RETRY MASK=WRITE
6107      024400 000406 BR EC2432 ; BR TO END 'C'
6108      024402 012737 000020 024700 LC2432: MOV #BIT04,RTMASK ; SET RETRY MASK=CRC
6109      024410 052737 000004 024700 BIS #BIT02,RTMASK ; SET RETRY MASK=READ
6110      024416 000240 EC2432: NOP ;
6111      024420 004737 024556 CALL SFERLG ; CALL SOFT ERROR LOGGER
6112      024424 000207 RETURN ; RETURN
6113      ;MOD 2.4.3.2 ----- END MODULE -----
    
```

```
6115 . SBTTL MOD 2.4.3.3 - UPDATE SOFT ERROR STATISTICS
6116 ;
6117 ;-----
6118 024426 000240 UDSFST: NOP ;
6119 024430 013702 024140 MOV CLASHD,R2 ; PUT CLASSIFICATION WORD IN R1
6120 024434 006202 ASR R2 ; SHIFT WORD RIGHT
6121 024436 006202 ASR R2 ; 3 TIMES TO GET
6122 024440 006202 ASR R2 ; RETRY COUNTER OFFSET (LAST 6 BITS)
6123 024442 042702 177700 BIC #177700,R2 ; CLEAR TOP 10 BITS
6124 024446 010237 024144 MOV R2,RTOFF ; SET RETRY COUNTER OFFSET
6125 024452 013702 024140 IA2433: MOV CLASHD,R2 ; GET CLASSIFICATION WORD
6126 024456 042702 177770 BIC #177770,R2 ; CLEAR ALL BIT ERROR CLASSIFICATION
6127 024462 022702 000006 CMP #6,R2 ; IF ERROR
6128 024466 001022 BNE LA2433 ; CLASS=DATA, THEN
6129 024470 032737 010000 024146 IB2433: BIT #BIT12,TSTCK ; IF TEST HAS
6130 024476 001404 BEQ LB2433 ; READ AFTER WRITE (RAW) BIT SET, THEN
6131 024500 012737 000012 024700 MOV #12,RTMASK ; SET DATA & WRITE RETRY
6132 024506 000403 BR EB2433 ; BR TO END IF 'B'
6133 024510 012737 000014 024700 LB2433: MOV #14,RTMASK ; SET DATA & READ RETRY
6134 024516 012737 000010 024144 EB2433: MOV #10,RTOFF ; SET DATA RT COUNTER OFFSET
6135 024524 012737 000050 024142 MOV #50,LOGOFF ; SET DATA LOG OFFSET
6136 024532 000405 BR EA2433 ; BR TO END 'A'
6137 024534 010237 024700 LA2433: MOV R2,RTMASK ; ELSE-PUT CLASS INTO RETRY MASK
6138 024540 162737 000050 024142 SUB #50,LOGOFF ; ADJ. LOG OFFSET SO THAT 'SEK' IS LOG BEGIN
6139 024546 004737 024556 EA2433: CALL SFERLG ; CALL SOFT ERROR LOGGER
6140 024552 000240 NOP ;
6141 024554 000207 X2433: RTS PC ; RETURN
6142 ;MOD 2.4.3.3 ----- END MODULE -----
```

```

6144          .SBTTL MOD 2.4.U1 - SOFT ERROR LOGGER
6145          ;
6146          ;-----
6147 024556 000240          SFERLG: NOP          ;
6148 024560 013701 024142          MOV          LOGOFF,R1          ;GET ERR LOG OFFSET
6149 024564 013702 024144          MOV          RTOFF,R2          ;GET RETRY COUNTER OFFSET
6150 024570 062702 002244          ADD          #SEEKRT,R2          ;CAL. RETRY COUNTER ADR
6151 024574 032737 000004 002164 IA24U1: BIT          #BIT02,SWREG          ;IF RETRY ON ERROR, LOG SOFT OR HD ERROR
6152 024602 001412          BEQ          LB24U1          ;SET (SFT SW REG), THEN
6153 024604 021227 000012          IB24U1: CMP          (R2),#12          ;IF RETRY COUNTER
6154 024610 103007          BHIS         LB24U1          ;EQUALS < 10 ERRORS, THEN
6155 024612 005212          INC          (R2)          ;INCREMENT RETRY COUNTER
6156 024614 053737 024700 002242          BIS          RTMASK,RETRY          ;SET RT FLAGS PER RT MASK
6157 024622 005037 002306          CLR          HARDER          ;CLEAR HARD ERROR
6158 024626 000413          BR          EB24U1          ;BR TO END 'B'
6159 024630 062701 004602          LB24U1: ADD          #HSEK,R1          ;HD ERR LOG ADR=HARD SEEK ADR+LOG OFFSET
6160 024634 063701 002220          ADD          UUTOFF,R1          ;UUT ERR LOG ADR=UUT OFFSET+LOG ADR
6161 024640 005211          INC          (R1)          ;INCREMENT UUT HARD ERROR LOG
6162 024642 043737 024700 002242          BIC          RTMASK,RETRY          ;CLEAR RETRY FLAGS USING RT MASK
6163 024650 005012          CLR          (R2)          ;CLEAR RETRY COUNTER
6164 024652 005237 002306          INC          HARDER          ;SET HARD ERROR FLAG
6165 024656 013701 024142          EB24U1: MOV          LOGOFF,R1          ;GET ERR LOG OFFSET
6166 024662 062701 004512          ADD          #SEK,R1          ;ERR LOG ADR=SEK LOG ADR+LOG OFFSET
6167 024666 063701 002220          ADD          UUTOFF,R1          ;UUT ERR LOG ADR=UUT OFFSET+LOG ADR
6168 024672 005211          INC          (R1)          ;INCREMENT UUT ERROR LOG
6169 024674 000240          X24U1: NOP          ;
6170 024676 000207          RTS          PC          ;RETURN
6171          ;
6172 024700 000000          RTMASK: 0          ;RETRY MASK
6173          ;MOD 2.4.U1 ----- END MODULE -----
6174          ;
  
```

```

6176 . SBTTL MOD 2.4.4 - EVALUATE UNIT ERROR CODE
6177 ;-----
6178
6179 024702 013701 002312 EVUTEC: MOV XERUUT,R1 ;GET ERR CODE & SAVE
6180 024706 042701 177400 BIC #177400,R1 ;CLEAR TOP BYTE
6181 024712 005701 IFA244: TST R1 ;IF ERRCODE
6182 024714 001443 BEQ END244 ;NOT=0, THEN
6183 024716 006201 ASR R1 ;SHIFT ERR CODE FOR LOOK UP
6184 024720 006201 ASR R1 ;AND ADDRESSING
6185 024722 062701 025030 ADD #ECCLAS,R1 ;CAL ERR TABLE CLASSIFICATION ADR
6186 024726 011102 MOV (R1),R2 ;GET ERR CODE CLASSIFICATION WORD
6187 024730 105702 IFB244: TSTB R2 ;IF LOWER BYTE
6188 024732 001003 BNE IFC244 ;EQUALS 0, THEN
6189 024734 050237 002232 BIS R2,SYSERR ;SET ERR ONTO SYSERR
6190 024740 000431 BR END244 ;BR TO END IF 'B'
6191 024742 122702 000300 IFC244: CMPB #300,R2 ;IF LOW BYTE
6192 024746 001024 BNE ELC244 ;EQUALS 300, THEN
6193 024750 022737 000003 025026 IFD244: CMP #3,FNEV4 ;IF FUNCTION WAS
6194 024756 001004 BNE IFE244 ;A READ, THEN
6195 024760 052737 002000 002234 BIS #2000,ERRTYP ;SET READ ERR
6196 024766 000416 BR END244 ;BR TO END IF 'B'
6197 024770 022737 000002 025026 IFE244: CMP #2,FNEV4 ;IF FUNCTION WAS
6198 024776 001004 BNE ELE244 ;A WRITE, THEN
6199 025000 052737 004000 002234 BIS #4000,ERRTYP ;SET WRITE ERROR
6200 025006 000406 BR END244 ;BR TO END IF 'B'
6201 025010 052737 040000 002234 ELE244: BIS #40000,ERRTYP ;SET UNK ERROR
6202 025016 000402 BR END244 ;BR TO END IF 'B'
6203 025020 050237 002234 ELC244: BIS R2,ERRTYP ;SET CLASSIFIED ERROR ONTO ERRTYP
6204 025024 000207 END244: RTS PC ;RETURN
6205 ;-----
6206 025026 000000 FNEV4: 0 ;FUNCTION FOR EVALUATION
6207 ;-----
6208 025030 000000 ECCLAS: .WORD 0 ;ERR CODE # 00 ---> NOT USED (NO ERROR)
6209 025032 000001 .WORD 1 ;ERR CODE # 10 ---> SEEK
6210 025034 000001 .WORD 1 ;ERR CODE # 20 ---> SEEK
6211 025036 000000 .WORD 0 ;ERR CODE # 30 ---> NOT ASSIGNED
6212 025040 004000 .WORD 4000 ;ERR CODE # 40 ---> SYS ERR
6213 025042 000001 .WORD 1 ;ERR CODE # 50 ---> SEEK
6214 025044 002000 .WORD 2000 ;ERR CODE # 60 ---> SELF DIAG ERR
6215 025046 000300 .WORD 300 ;ERR CODE # 70 ---> READ OR WRITE ERR
6216 025050 004000 .WORD 4000 ;ERR CODE # 100 ---> SYS ERR
6217 025052 000300 .WORD 300 ;ERR CODE # 110 ---> READ OR WRITE ERR
6218 025054 000300 .WORD 300 ;ERR CODE # 120 ---> READ OR WRITE ERR
6219 025056 000300 .WORD 300 ;ERR CODE # 130 ---> READ OR WRITE ERR
6220 025060 000002 .WORD 2 ;ERR CODE # 140 ---> CRC ERR
6221 025062 000001 .WORD 1 ;ERR CODE # 150 ---> SEEK ERR
6222 025064 000300 .WORD 300 ;ERR CODE # 160 ---> READ OR WRITE ERR
6223 025066 000300 .WORD 300 ;ERR CODE # 170 ---> READ OR WRITE ERR
6224 025070 000002 .WORD 2 ;ERR CODE # 200 ---> CRC ERR
6225 025072 000000 .WORD 0 ;ERR CODE # 210 ---> NOT ASSIGNED
6226 025074 002000 .WORD 2000 ;ERR CODE # 220 ---> SELF DIAG ERR
6227 025076 004000 .WORD 4000 ;ERR CODE # 230 ---> SYS ERR
6228 025100 020000 .WORD 20000 ;ERR CODE # 240 ---> DENSITY ERR
6229 025102 020000 .WORD 20000 ;ERR CODE # 250 ---> DENSITY ERR
6230 025104 000000 .WORD 0 ;ERR CODE # 260 ---> NOT ASSIGNED
6231 ;MOD 2.4.4 ----- END MODULE -----
    
```

```

6233
6234          . SBTTL MOD 2.5 - OUTPUT ERROR TYPE
6235          ;-----
6236
6237 025106 013737 002234 030044 OTERTP: MOV      ERRTP,ERRREG ;SET ERROR TYPE FOR PRINT OUT
6238 025114 000240          NOP          ;
6239 025116 013701 002234          MOV      ERRTP,R1      ;GET ERROR TYPE
6240 025122 005002          CLR      R2          ;CLEAR ERROR # COUNT
6241 025124 000240          BDA25:  NOP          ;
6242 025126 032701 000001          IFA25:  BIT      #1,R1      ;IF BIT #1
6243 025132 001405          BEQ      ELA25      ;EQUALS 1, THEN
6244 025134 010204          MOV      R2,R4      ;SAVE ERROR # COUNT
6245 025136 006304          ASL      R4          ;DOUBLE ERR # COUNT FOR ADDRESSING
6246 025140 062704 025760          ADD      #ET1,R4     ;SET ADDR FOR ERR MSG PRINT
6247 025144 000407          BR      THA25      ;BR TO THEN 'A'
6248 025146 000241          ELA25:  CLC          ;CLEAR CARRY BIT
6249 025150 006201          ASR      R1          ;SHIFT ERR TYPE RIGHT
6250 025152 005202          INC      R2          ;INCREMENT ERROR # COUNT
6251 025154 022702 000017          CMP      #17,R2     ;DO UNTIL ERROR # COUNT
6252 025160 001361          BNE     BDA25      ;EQUALS 15, THEN
6253 025162 000464          BR      EIA25      ;BR TO END IF 'A'
6254 025164 005003          THA25:  CLR      R3          ;CLEAR R3
6255 025166 010205          MOV      R2,R5      ;GET ERR#
6256 025170 062705 026020          ADD      #ETCLAS,R5 ;CAL. ERR# CLASSIFICATION ADR
6257 025174 111503          MOV     (R5),R3     ;GET ERR# CLASSIFICATION
6258 025176 032703 000001          IFB25:  BIT      #1,R3      ;IF SOFT ERR
6259 025202 001413          BEQ      IFC25      ;CLASS, THEN
6260 025204 005737 002306          TST     HARDER     ;IF HARD ERR
6261 025210 001013          BNE     ELB25      ;NOT SET, THEN
6262 025212 010237 025224          MOV      R2,SFTE1+2 ;SET ERR #
6263 025216 011437 025226          MOV      (R4),SFTE1+4 ;SET ERR MSG
6264 025222          SFTE1:  ERRSOFT 0,0 ;SOFT ERROR
6265 025230 000416          BR      EIC25      ;
6266 025232 032703 000002          IFC25:  BIT      #2,R3      ;IF HARD ERR
6267 025236 001413          BEQ      EIC25      ;CLASS, THEN
6268 025240 052702 000040          ELB25:  BIS      #40,R2     ;SET HARD ERROR #
6269 025244 010237 025256          MOV      R2,HDTE1+2 ;SET ERR #
6270 025250 011437 025260          MOV      (R4),HDTE1+4 ;SET ERR MSG
6271 025254          HDTE1:  ERRHRD 0,0 ;HARD ERROR
6272 025262 005237 002306          INC     HARDER     ;SET HARD ERROR FLAG
6273 025266 013737 002234 030044          EIC25:  MOV      ERRTP,ERRREG ;SET ERR TYPE FOR PRINT OUT
6274 025274 004737 027644          CALL    PRERR      ;CALL U.P. ERR - PRINT ERR INFO
6275 025300 013737 002234 015670          MOV      ERRTP,ERTSAV ;SAVE ERR TYP FOR DATA CK
6276 025306 005037 002234          CLR     ERRTP      ;CLEAR DEVICE ERR
6277 025312 004737 030274          CALL    XERPRT     ;CALL MOD U.PRT.B - PRINT ERR CODE
6278 025316 005737 002306          IFD25:  TST     HARDER     ;IF NOT A
6279 025322 001002          BNE     ELD25      ;HARDER, THEN
6280 025324 004737 026040          CALL    PTRTY      ;CALL 2.5.1 - PRINT RETRY #
6281 025330 005037 002306          ELD25:  CLR     HARDER     ;CLEAR HARD ERROR FLAG
6282 025334 000240          EIA25:  NOP          ;
6283 025336 000207          RTS     PC          ;RETURN
6284          ;-----
    
```



```

6286
6287 025340 051440 042505 020113 ;-----
6288 025352 041440 041522 042440 ERT1: .ASCIZ / SEEK ERR/
6289 025363 040 045503 051440 ERT2: .ASCIZ / CRC ERR/
6290 025377 040 040504 040524 ERT3: .ASCIZ / CK SUM ERR/
6291 025411 040 047125 051501 ERT4: .ASCIZ / DATA ERR/
6292 025425 040 042504 027114 ERT5: .ASCIZ / UNASSG ERR/
6293 025457 040 042504 027114 ERT6: .ASCIZ / DEL. DATA UNEXPECTED ERR/
6294 025506 052440 040516 051523 ERT7: .ASCIZ / DEL. DATA MISSING ERR/
6295 025522 052440 045516 042440 ERT8: .ASCIZ / UNASSG ERR/
6296 025533 040 044506 046114 ERT9: .ASCIZ / UNK ERR/
6297 025565 040 042522 042101 ERT10: .ASCIZ / FILL OR EMPTY BUFFER ERR/
6298 025577 040 051127 052111 ERT11: .ASCIZ / READ ERR/
6299 025612 044440 052116 051105 ERT12: .ASCIZ / WRITE ERR/
6300 025651 040 047504 042516 ERT13: .ASCIZ / INTERRUPT BUT NO DONE BIT ERR/
6301 025710 042440 051122 051117 ERT14: .ASCIZ / DONE BIT BUT NO INTERRUPT ERR/
6302 025743 040 051105 020122 ERT15: .ASCIZ / ERROR, BUT NO ERR BIT SET/
6303 ERT16: .ASCIZ / ERR BIT SET/

```

```

6304 025760 025340 ET1: .EVEN
6305 025762 025352 .WORD ERT1
6306 025764 025363 .WORD ERT2
6307 025766 025377 .WORD ERT3
6308 025770 025411 .WORD ERT4
6309 025772 025425 .WORD ERT5
6310 025774 025457 .WORD ERT6
6311 025776 025506 .WORD ERT7
6312 026000 025522 .WORD ERT8
6313 026002 025533 .WORD ERT9
6314 026004 025565 .WORD ERT10
6315 026006 025577 .WORD ERT11
6316 026010 025612 .WORD ERT12
6317 026012 025651 .WORD ERT13
6318 026014 025710 .WORD ERT14
6319 026016 025743 .WORD ERT15
6320 .WORD ERT16

```

```

6321 ;-----
6322 ; ERROR - TYPE - ERR#
6323 026020 001 ETCLAS: .BYTE 1 ;SEEK - SOFT - 0 -32
6324 026021 001 .BYTE 1 ;CRC - SOFT - 1 -33
6325 026022 002 .BYTE 2 ;CKSUM - HARD - -34
6326 026023 001 .BYTE 1 ;DATA - SOFT - 3 -35
6327 026024 000 .BYTE 0 ;UNASSIGNED -
6328 026025 002 .BYTE 2 ;DEL. DATA UNEX - HARD - -37
6329 026026 002 .BYTE 2 ;DEL. DATA MISSING - HARD - -38
6330 026027 000 .BYTE 0 ;UNASSIGNED -
6331 026030 002 .BYTE 2 ;UNK ERR - HARD - -40
6332 026031 002 .BYTE 2 ;FILL/EMPTY BUFFER - HARD - -41
6333 026032 001 .BYTE 1 ;READ - SOFT - 10-42
6334 026033 001 .BYTE 1 ;WRITE - SOFT - 11-43
6335 026034 002 .BYTE 2 ;INTER-BUT NO DONE - HARD - -44
6336 026035 002 .BYTE 2 ;DONE-BUT NO INTER - HARD - -45
6337 026036 002 .BYTE 2 ;ERR-BUT NO ERR BIT - HARD - -46
6338 026037 002 .BYTE 2 ;ERR BIT SET - HARD - -47
6339 .EVEN

```

;MOD 2.5 ----- END MODULE -----

6340
6341

6343
 6344
 6345
 6346
 6347
 6348
 6349
 6350
 6351
 6352
 6353
 6354
 6355
 6356
 6357
 6358
 6359
 6360
 6361
 6362
 6363
 6364
 6365
 6366
 6367
 6368
 6369
 6370
 6371
 6372
 6373
 6374
 6375
 6376
 6377
 6378
 6379
 6380
 6381
 6382
 6383
 6384
 6385
 6386
 6387
 6388
 6389
 6390
 6391
 6392
 6393
 6394
 6395
 6396
 6397
 6398

.SBTTL MOD 2.5.1 - PRINT RETRY

```

;-----
PTRTY:  NOP
IFA251: TST    RETRY      ; IF RETRY
        BEQ    END251     ; NOT=0, THEN
IFB251: BIT    #1,RETRY   ; IF RETRY
        BEQ    IFC251     ; IS SEEK, THEN
        MOV    SEEKRT,R1  ; SET SEEK RT COUNT
        MOV    #MSKRT,R2  ; SET SEEK RT MSG
        BR     EIB251     ; BR TO END IF 'B'
IFC251: BIT    #2,RETRY   ; IF RETRY
        BEQ    IFE251     ; IS WRT, THEN
IFD251: BIT    #30,RETRY  ; IF RETRY
        BEQ    ELD251     ; IS DATA OR CRC, THEN
IFG251: BIT    #10,RETRY  ; IF RETRY
        BEQ    ELG251     ; IS DATA, THEN
        MOV    DATART,R1  ; SET DATA RT COUNT
        MOV    #MDWTRT,R2 ; SET DATA WRT MSG
        BR     EIB251     ; BR TO END IF 'B'
ELG251: MOV    CRCRT,R1   ; SET CRC RETRY COUNT
        MOV    #MCWTRT,R2 ; SET CRC WRT MSG
        BR     EIB251     ; BR TO END IF 'B'
ELD251: MOV    WTRT,R1    ; SET WRT RT COUNT
        MOV    #MWTRT,R2  ; SET WRT RT MSG
        BR     EIB251     ; BR TO END IF 'B'
IFE251: BIT    #4,RETRY   ; IF RETRY
        BEQ    END251     ; IS READ, THEN
IFF251: BIT    #30,RETRY  ; IF RETRY
        BEQ    ELF251     ; IS DATA OR CRC, THEN
IFH251: BIT    #10,RETRY  ; IF RETRY
        BEQ    ELH251     ; IS DATA, THEN
        MOV    DATART,R1  ; SET DATA RT COUNT
        MOV    #MORDRT,R2 ; SET DATA READ RT MSG
        BR     EIB251     ; BR TO END IF 'B'
ELH251: MOV    CRCRT,R1   ; SET CRC RETRY COUNT
        MOV    #MCRDRT,R2 ; SET CRC READ MSG
        BR     EIB251     ; BR TO END IF 'B'
ELF251: MOV    READRT,R1  ; SET READ RT COUNT
        MOV    #MRDRT,R2  ; SET READ RT MSG
        NOP
EIB251: PRINTB R2,R1      ; PRINT RETRY # & TYPE
        NOP
END251: RTS    PC        ; RETURN
;-----
MSKRT:  .ASCIZ  /%A SEEK RETRY#%D2%N/
MDWTRT: .ASCIZ  /%A DATA WRITE RETRY#%D2%N/
MWTRT:  .ASCIZ  /%A WRITE RETRY#%D2%N/
MORDRT: .ASCIZ  /%A DATA READ RETRY#%D2%N/
MRDRT:  .ASCIZ  /%A READ RETRY#%D2%N/
MCWTRT: .ASCIZ  /%A CRC WRITE RETRY#%D2%N/
MCRDRT: .ASCIZ  /%A CRC READ RETRY#%D2%N/
        .EVEN
;MOD 2.5.1 ----- END MODULE -----

```

```

6400          . SBTTL MOD 2.6 - SET DRIVES DONE
6401          ;-----
6402
6403 026534 000240          STDVDN: NOP          ;
6404 026536 005737 015660  IFA26: TST          DVDNCK          ; IF DRV DONE CK
6405 026542 001430          BEQ          END26          ; IS SET, THEN
6406 026544 000240          NOP          ;
6407 026546 005037 015660  CLR          DVDNCK          ; CLEAR DRV DONE CK
6408 026552 032737 000001 002214  IFB26: BIT          #1,UUT          ; IF DRV#1 DONE
6409 026560 001404          BEQ          ELB26          ; THEN
6410 026562 052737 000002 015650  BIS          #2,BTHDRV          ; SET DRV#1 DONE FLAG
6411 026570 000403          BR          EIB26          ; BR TO END
6412 026572 052737 000001 015650  ELB26: BIS          #1,BTHDRV          ; SET DRV#0 DONE FLAG
6413 026600 005001          EIB26: CLR          R1          ; CLEAR TEMP DRV DONE REG
6414 026602 013703 002214  MOV          UUT,R3          ; GET UNIT UNDER TEST
6415 026606 000261          SEC          ; SET CARRY BIT
6416 026610 006101          BDA26: ROL          R1          ; MOVE DRV BIT
6417 026612 005303          DEC          R3          ; DECREMENT UNIT UNDER TEST
6418 026614 005703          TST          R3          ; DO UNTIL UNIT UNDER TST
6419 026616 002374          DUA26: BGE          BDA26          ; EQUALS -1
6420 026620 050137 015662  BIS          R1,DRVDN          ; THEN SET THIS DRV DONE
6421 026624 000207          END26: RTS          PC          ; RETURN
6422          ;MOD 2.6 -----END MODULE-----
    
```

6424
6425
6426 026626 000240
6427 026630 023737 002212 002210
6428 026636 001003
6429 026640 012737 000001 010244
6430 026646 000207
6431

```
.SBTTL MOD 3.0 - OUTPUT EXERCISE COMPLETE  
;  
-----  
OTEXCM: NOP  
          CMP      SUT,SDD      ; IF ALL SCHEDULED  
          BNE     END30        ; DRIVE DONE  
          MOV     #1,EXCMP     ; SET EXERCISE COMPLETE  
END30:   RTS      PC          ; RETURN  
;MOD 3.0 ----- END MODULE -----
```

6433
6434
6435 026650 000240
6436 026652 013701 002232
6437 026656 000241
6438 026660 006201
6439 026662 000241
6440 026664 006201
6441 026666 006201
6442 026670 005002
6443 026672 000240
6444 026674 032701 000001
6445 026700 001405
6446 026702 010204
6447 026704 006304
6448 026706 062704 027574
6449 026712 000406
6450 026714 006201
6451 026716 005202
6452 026720 022702 000017
6453 026724 001362
6454 026726 000447
6455 026730 010205
6456 026732 062705 027626
6457 026736 111503
6458 026740 032703 000002
6459 026744 001413
6460 026746 010205
6461 026750 052705 000100
6462 026754 010537 026766
6463 026760 011437 026770
6464 026764
6465 026772 000415
6466 026774 032703 000004
6467 027000 001412
6468 027002 010205
6469 027004 052705 000200
6470 027010 010537 027022
6471 027014 011437 027024
6472 027020
6473 027026 000240
6474 027030 013737 002232 030044
6475 027036 004737 027644
6476 027042 004737 030274
6477 027046 000240
6478 027050 005037 002232
6479 027054 000207
6480
6481

```
.SBTTL MOD 4.0 - OUTPUT SYSTEM ERROR
-----
OTSYER: NOP
MOV SYSERR,R1 ;GET SYSTEM ERR
CLC ;CLEAR CARRY BIT
ASR R1 ;SHIFT
CLC
ASR R1 ;FUNCTION
ASR R1 ;OUT
CLR R2 ;CLEAR ERR # COUNT
BDA40: NOP
IFA40: BIT #1,R1 ;IF BIT #1
BEQ ELA40 ;EQUALS 1, THEN
MOV R2,R4 ;SAVE ERROR # COUNT
ASL R4 ;DOUBLE ERR # COUNT FOR ADDRESSING
ADD #SE1,R4 ;SET ADDR FOR ERR MSG PRINT
BR THA40 ;BR TO THEN 'A'
ELA40: ASR R1 ;SHIFT ERR TYPE RIGHT
INC R2 ;INCREMENT ERROR # COUNT
CMP #17,R2 ;DO UNTIL ERR # COUNT
BNE BDA40 ;EQUALS 15, THEN
BR EIA40 ;BR TO END IF 'A'
THA40: MOV R2,R5 ;GET ERR#
ADD #ESCLAS,R5 ;CAL. ERR# CLASSIFICATION ADR
MOVB (R5),R3 ;GET ERR# CLASSIFICATION
IFB40: BIT #2,R3 ;IF DEVICE FATAL
BEQ IFC40 ;ERROR, THEN
MOV R2,R5 ;GET ERR#
BIS #100,R5 ;SET ERR CLASS=SYS
MOV R5,DVFER1+2 ;SET ERR#
MOV (R4),DVFER1+4 ;SET ERR MSG
DVFER1: ERDF 0,0
BR EIC40 ;BR TO END IF 'C'
IFC40: BIT #4,R3 ;IF SYSTEM FATAL
BEQ EIC40 ;ERROR, THEN
MOV R2,R5 ;GET ERR#
BIS #200,R5 ;SET ERR CLASS=SYS
MOV R5,SYFER1+2 ;SET ERR#
MOV (R4),SYFER1+4 ;SET ERR MSG
SYFER1: ERSF 0,0
EIC40: NOP
MOV SYSERR,ERRREG ;SET SYS ERR FOR PRINT OUT
CALL PRERR ;CALL U.P. ERR - PRINT ERR INFO
CALL XERPRT ;CALL MOD U.PRT. B - PRINT ERROR CODE
EIA40: NOP
CLR SYSERR ;CLEAR SYS ERRORS
END40: RTS PC
-----
```

```

6483
6484
6485 027056 047040 020117 047504 SYSE4: .ASCIZ / NO DONE BIT ON INITIALIZE/
6486 027111 040 047516 042040 SYSE5: .ASCIZ / NO DONE BIT ON FUNCTION/
6487 027142 047040 020117 051104 SYSE6: .ASCIZ / NO DRIVE READY BIT/
6488 027166 047040 020117 044523 SYSE7: .ASCIZ / NO SIDE READY BIT/
6489 027211 040 047516 042040 SYSE8: .ASCIZ / NO DONE BIT AFTER READ STATUS/
6490 027250 053440 047522 043516 SYSE9: .ASCIZ / WRONG DRIVE RESPONDING/
6491 027300 053440 047522 043516 SYSE10: .ASCIZ / WRONG SIDE RESPONDING/
6492 027327 040 047125 051525 SYSE11: .ASCIZ / UNUSED/
6493 027337 040 047125 051525 SYSE12: .ASCIZ / UNUSED/
6494 027347 040 044504 045523 SYSE13: .ASCIZ / DISKETTE WRONG DENSITY ERR/
6495 027403 040 042504 051516 SYSE14: .ASCIZ / DENSITY ERR/
6496 027420 052040 046511 020105 SYSE15: .ASCIZ / TIME OUT ON "TR" OR "DONE" BIT/
6497 027460 052440 041516 040514 SYSE16: .ASCIZ / UNCLASSIFIED SYSTEM ERROR/
6498 027513 045 022516 043101 FUNCT: .ASCIZ /%N%AFUNCTION CODE:%03/
6499 027541 045 022516 051501 ERRORS: .ASCIZ /%N%ASYSTEM ERROR REG=%B%N/
6500
6501 027574 027056 SE1: .EVEN
6502 027576 027111 .WORD SYSE4
6503 027600 027142 .WORD SYSE5
6504 027602 027166 .WORD SYSE6
6505 027604 027211 .WORD SYSE7
6506 027606 027250 .WORD SYSE8
6507 027610 027300 .WORD SYSE9
6508 027612 027327 .WORD SYSE10
6509 027614 027337 .WORD SYSE11
6510 027616 027347 .WORD SYSE12
6511 027620 027403 .WORD SYSE13
6512 027622 027420 .WORD SYSE14
6513 027624 027460 .WORD SYSE15
6514
6515
6516
6517 027626 004 ESCLAS: .BYTE 4 ;ERROR - CLASS -ERR#
6518 027627 002 .BYTE 2 ;-----
6519 027630 002 .BYTE 2 ;NO DONE ON INIT - SYS FATAL - 128
6520 027631 002 .BYTE 2 ;NO DONE ON FUNCTION - DEV FATAL - 65
6521 027632 004 .BYTE 4 ;NO DRIVE RDY - DEV FATAL - 66
6522 027633 004 .BYTE 4 ;NO SIDE RDY - DEV FATAL - 67
6523 027634 000 .BYTE 0 ;NO DONE AFTER RD STA - DEV FATAL - 68
6524 027635 000 .BYTE 0 ;WRG DRV RESPOND - SYS FATAL - 133
6525 027636 002 .BYTE 2 ;WRG SIDE RESPOND - SYS FATAL - 134
6526 027637 002 .BYTE 2 ;UNUSED - 0
6527 027640 004 .BYTE 4 ;UNUSED - 0
6528 027641 004 .BYTE 4 ;DISKETT WRG DEN - DEV FATAL - 73
6529 027642 004 .BYTE 4 ;DENSITY ERR - DEV FATAL - 74
6530 027644 .BYTE 4 ;T. O. ON "TR" OR "DONE" - SYS FATAL - 139
6531 .EVEN .BYTE 4 ;SYS ERR - SYS FATAL - 140

;MOD 4.0 ----- END MODULE -----
    
```

6533
6534
6535
6536
6537 027644
6538 027704 005737 002310
6539 027710 001452
6540 027712
6541 027762
6542 030032 005037 002310
6543 030036 005037 030044
6544 030042 000207
6545
6546 030044 000000
6547
6548 030046 040445 052440 044516
6549 030123 045 020101 051105
6550 030204 040445 052040 051124
6551 030274
6552
6553

.SBTTL MOD U. PRT. ERR - PRINT ERRORS

```
-----  
PRERR: PRINTB #IDENT1,UNIT,CSRUIT,ESRUIT,MDOT;  
IFAUP: TST PRTECD ;IF ERR CODE FLAG  
BEQ ENDUP ;SET, THEN  
PRINTX #XER1,<B,XERUIT>,<B,WC>,<B,CTKO>,<B,CTK1>  
PRINTX #XER2,<B,TTRK>,<B,TSEC>,<B,SFTSTS>,<B,BTRK>  
CLR PRTECD ;CLEAR ERR CODE FLAG  
ENDUP: CLR ERRREG ;CLEAR ERR REGISTER  
RTS PC ;RETURN  
-----  
ERRREG: 0 ;  
-----  
IDENT1: .ASCIZ /%A UNIT#%01%A RXCSR=%0%A RXESR=%0%A CMD=%0%A/  
XER1: .ASCIZ /%A ERCD=%03%A WC=%03%A CTRKO=%D2%A CTRK1=%D2%A/  
XER2: .ASCIZ /%A TTRK=%D2%A TSEC=%D2%A SFTSTAT=%03%A BTRK=%D2%A %N/  
.EVEN  
;MOD U. PRT. ERR ----- END MODULE -----
```

6555
 6556
 6557
 6558
 6559 030274 105737 002312
 6560 030300 001425
 6561 030302 013701 002312
 6562 030306 042701 177400
 6563 030312 006201
 6564 030314 006201
 6565 030316 062701 030356
 6566 030322 011137 030356
 6567 030326 000240
 6568 030330
 6569 030350 105037 002312
 6570 030354 000207
 6571
 6572
 6573 030356 000000
 6574
 6575
 6576 030360 030432
 6577 030362 030514
 6578 030364 030576
 6579 030366 030633
 6580 030370 030714
 6581 030372 031005
 6582 030374 031044
 6583 030376 031142
 6584 030400 031236
 6585 030402 031326
 6586 030404 031375
 6587 030406 031500
 6588 030410 031602
 6589 030412 031706
 6590 030414 031752
 6591 030416 032025
 6592 030420 032104
 6593 030422 032141
 6594 030424 032227
 6595 030426 032264
 6596 030430 032313
 6597

```

.SBTTL MOD U.PRT. EC - PRINT UNIT ERROR CODE
;-----
XERPRT: TSTB XERUUT ; IF ERROR
        BEQ ENDXER ; NOT=0, THEN
        MOV XERUUT,R1 ; SAVE EXTENDED ERR CODE IN TEMP #1
        BIC #177400,R1 ; CLR TOP BYTE
        ASR R1 ; FORMAT E. C.
        ASR R1 ; FORMAT E. C. FOR ADR
        ADD #ECTAB-2,R1 ; FIND ADR OF ERROR MSG
        MOV (R1),EXMSG ; SET ADR OF ERROR MSG FOR PRINT
        NOP ;
        PRINTX EXMSG ; PRINT UNIT CODE ERROR MSG
        CLRB XERUUT ; CLEAR ERROR CODE
ENDXER: RTS PC ; RETURN
;-----
EXMSG: 0 ; MSG ADR FOR PRINT
;-----
ECTAB: .WORD EC1
        .WORD EC2
        .WORD EC3
        .WORD EC4
        .WORD EC5
        .WORD EC6
        .WORD EC7
        .WORD EC10
        .WORD EC11
        .WORD EC12
        .WORD EC13
        .WORD EC14
        .WORD EC15
        .WORD EC16
        .WORD EC17
        .WORD EC20
        .WORD EC21
        .WORD EC22
        .WORD EC23
        .WORD EC24
        .WORD EC25
;-----

```



```
6599 ;-----  
6600 030432 040445 020040 037040 EC1: .ASCIZ /XA >DRIVE 0 FAILED TO SEE HOME ON INITIALIZE. %N/  
6601 030514 040445 020040 037040 EC2: .ASCIZ /XA >DRIVE 1 FAILED TO SEE HOME ON INITIALIZE. %N/  
6602 030576 040445 020040 037040 EC3: .ASCIZ /XA >UNASSIGNED ERR CODE. %N/  
6603 030633 045 020101 020040 EC4: .ASCIZ /XA >TRIED TO ACCESS A TRACK GREATER THAN 76. %N/  
6604 030714 040445 020040 037040 EC5: .ASCIZ /XA >HOME WAS FOUND BEFORE DESIRED TRACK WAS REACHED. %N/  
6605 031005 045 020101 020040 EC6: .ASCIZ /XA >SELF DIAGNOSTIC ERROR. %N/  
6606 031044 040445 020040 037040 EC7: .ASCIZ /XA >DESIRED SECTOR NOT FOUND AFTER LOOKING AT 52 HEADERS. %N/  
6607 031142 040445 020040 037040 EC10: .ASCIZ /XA >WRITE FUNCTION ATTEMPTED ON A WRITE PROTECTED DISK. %N/  
6608 031236 040445 020040 037040 EC11: .ASCIZ /XA >MORE THAN 40 MICROSECONDS AND NO SEPCLOCK SEEN. %N/  
6609 031326 040445 020040 037040 EC12: .ASCIZ /XA >A PREAMBLE COULD NOT BE FOUND. %N/  
6610 031375 045 020101 020040 EC13: .ASCIZ /XA >PREAMBLE FOUND BUT NO ID MARK FOUND WITHIN ALLOWABLE TIME. %N/  
6611 031500 040445 020040 037040 EC14: .ASCIZ /XA >CRC ERROR ON WHAT APPEARED TO BE A HEADER. ERROR NOT SET. %N/  
6612 031602 040445 020040 037040 EC15: .ASCIZ /XA >TRACK ADDRESS OF GOOD HEADER DOES NOT COMPARE WITH DESIRED. %N/  
6613 031706 040445 020040 037040 EC16: .ASCIZ /XA >TOO MANY TRIES FOR AN IDAM. %N/  
6614 031752 040445 020040 037040 EC17: .ASCIZ /XA >DATA AM NOT FOUND IN ALLOTTED TIME. %N/  
6615 032025 045 020101 020040 EC20: .ASCIZ /XA >CRC ERROR ON READING SECTOR FROM DISK. %N/  
6616 032104 040445 020040 037040 EC21: .ASCIZ /XA >UNASSIGNED ERR CODE. %N/  
6617 032141 045 020101 020040 EC22: .ASCIZ /XA >R-W ELECTRONICS FAILED MAINTENANCE MODE TEST. %N/  
6618 032227 045 020101 020040 EC23: .ASCIZ /XA >WORD COUNT OVERFLOW. %N/  
6619 032264 040445 020040 037040 EC24: .ASCIZ /XA >DENSITY ERROR. %N/  
6620 032313 045 020101 020040 EC25: .ASCIZ /XA >WRONG KEY WORD FOR SET DENSITY COMMAND. %N/  
6621 ;-----  
6622 032374 . EVEN
```

```

6624 .SBTTL MOD U. INTR. 1 - INTERRUPT HANDLER #0
6625 ;-----
6626 032374 000240 INTMO: NOP ;
6627 032376 013737 002226 032460 MOV UOADR, INCSAD ; SET UNIT #0 ADDRESS
6628 032404 004737 032430 CALL SVUTRG ; CALL MOD U. INTR. U - SAVE UNIT REG
6629 032410 000002 RTI ;
6630 ;MOD U. INTR. 1 ----- END MODULE -----
    
```

```

6631
6632
6633 .SBTTL MOD U. INTR. 2 - INTERRUPT HANDLER #1
6634 ;-----
6635
6636 032412 000240 INTM1: NOP ;
6637 032414 013737 002230 032460 MOV U1ADR, INCSAD ; SET UNIT #1 ADDRESS
6638 032422 004737 032430 CALL SVUTRG ; CALL MOD U. INTR. U - SAVE UNIT REG
6639 032426 000002 RTI ;
6640 ;MOD U. INTR. 2 ----- END MODULE -----
    
```

```

6641
6642
6643 .SBTTL MOD U. INTR. U - SAVE UNIT REG
6644 ;-----
6645
6646
6647 032430 000240 SVUTRG: NOP ;
6648 032432 012737 000001 021446 MOV #1, DNFLAG ; SET DONE FLAG
6649 032440 013701 032460 MOV INCSAD, R1 ; SAVE UUT ADDRESS
6650 032444 012137 002236 MOV (R1)+, CSRUUT ; SAVE UUT CSR
6651 032450 011137 002240 MOV (R1), ESRUUT ; SAVE UUT ESR
6652 032454 000240 NOP ;
6653 032456 000207 RTS PC ; RETURN
6654 ;-----
6655 032460 000000 INCSAD: 0 ; INTERRUPTING UNIT CSR ADDRESS
6656 ;MOD U. I. U ----- END MODULE -----
    
```

```

6657
6658
6659 .EVEN
6660
6661
    
```

6663
6674
6675
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6735
6736
6737
6738
6739
6740
6741
6742
6749

032462
032464
032474
032504
032516
032530
032532

054122 040440 042104
126 041505 047524
051104 053111 020105
054105 040520 051516

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

BGNHRD
GPRMA MSG1,0,0,0,177777,YES
GPRMA MSG2,2,0,0,177777,YES
GPRMD MSG3,4,0,177777,0,1,YES
GPRMD MSG4,6,0,177777,0,1,YES

EXIT HRD

ENDHRD

MSG1: .ASCIZ /RX ADDRESS/
MSG2: .ASCIZ /VECTOR ADDRESS/
MSG3: .ASCIZ /DRIVE #/
MSG4: .ASCIZ /EXPANSION-TYPE <CR>/

. EVEN

6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763 032620
6764
6765 032622
6766 032630
6767 032632
6768 032640
6769 032652
6770 032664
6771 032676
6772 032704
6773 032712
6774 032720
6775 032722
6776 032730
6777 032736
6778 032744
6779 032752
6780 032760
6781 032762
6782 032774
6783 033006
6784 033014
6785 033016
6786 033030
6787 033042
6788 033050
6789
6796
6797
6798
6799 033052
6800

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

          BGNSFT

          GPRML  MSG6, 2, 1, YES
          XFERF  1$
          GPRML  MSG7, 2, 2, YES
1$:      GPRMD  MSG8, 4, 0, 177777, 0, 6, YES
          GPRMD  MSG11, 6, 0, 177777, 0, 6, YES
          GPRMD  MSG14, 10, 0, 177777, 0, 6, YES
          GPRML  MSG15, 12, 1, YES
          GPRML  MSG16, 12, 2, YES
          GPRML  MSG17, 2, 100, YES
          XFERF  4$
          GPRML  MSG18, 12, 4, YES
          GPRML  MSG19, 12, 10, YES
          GPRML  MSG20, 12, 20, YES
          GPRML  MSG21, 12, 40, YES
4$:      GPRML  MSG22, 2, 200, YES
          XFERF  5$
          GPRMD  MSG23, 14, 0, 177777, 0, 76, YES
          GPRMD  MSG24, 16, 0, 177777, 0, 76, YES
5$:      GPRML  MSG25, 2, 400, YES
          XFERF  6$
          GPRMD  MSG26, 20, 0, 177777, 1, 26, YES
          GPRMD  MSG27, 22, 0, 177777, 1, 26, YES
6$:      GPRML  MSG5, 0, 177777, YES
          EXIT   SFT

          .EVEN

          ENDSFT
```

```

6802 ;-----
6803      000015      CR==15      ;CARRIAGE RETURN
6804      000012      LF==12      ;LINE FEED
6805 033052 052506 052524 042522 MSG5: .ASCIZ /FUTURE EXPANSION TYPE <CR> /
6806 033106 042524 052123 051440 MSG6: .ASCIZ /TEST SETUP HELP /
6807 033127      105 042530 041522 MSG7: .ASCII /EXERCISE OPTIONS/<CR><LF>
6808 033151      040 020040 020060 .ASCII / 0 = WRITE-READ-DATA CK & READ-DATA CK/<CR><LF>
6809 033223      040 020040 020061 .ASCII / 1 = WRITE ONLY/<CR><LF>
6810 033246 020040 031040 036440 .ASCII / 2 = WRITE-READ/<CR><LF>
6811 033271      040 020040 020063 .ASCII / 3 = WRITE-READ-DATA CHECK/<CR><LF>
6812 033327      040 020040 020064 .ASCII / 4 = READ-DATA CHECK ONLY/<CR><LF>
6813 033364 020040 032440 036440 .ASCII / 5 = READ ONLY (CRC CHECK)/<CR><LF>
6814 033422 020040 033040 036440 .ASCII / 6 = WRITE-READ-DATA CHECK ON ALTERNATE DRIVES/<CR><LF>
6815 033504 040504 040524 050040 .ASCII /DATA PATTERN OPTIONS/<CR><LF>
6816 033532 020040 030040 036440 .ASCII / 0 = RANDOM/<CR><LF>
6817 033551      040 020040 020061 .ASCII / 1 = ZEROS/<CR><LF>
6818 033567      040 020040 020062 .ASCII / 2 = ONES/<CR><LF>
6819 033604 020040 031440 036440 .ASCII / 3 = FLOATING ZERO/<CR><LF>
6820 033632 020040 032040 036440 .ASCII / 4 = FLOATING ONE/<CR><LF>
6821 033657      040 020040 020065 .ASCII / 5 = 125/<CR><LF>
6822 033673      040 020040 020066 .ASCII / 6 = 333/<CR><LF>
6823 033707      124 040522 045503 .ASCII /TRACK SEQUENCE OPTIONS/<CR><LF>
6824 033737      040 020040 020060 .ASCII / 0 = RANDOM/<CR><LF>
6825 033756 020040 030440 036440 .ASCII / 1 = INCREMENT O. D. /<CR><LF>
6826 034005      040 020040 020062 .ASCII / 2 = DECREMENT I. D. /<CR><LF>
6827 034034 020040 031440 036440 .ASCII / 3 = INCREMENT O. D. -DECREMENT I. D. /<CR><LF>
6828 034102 020040 032040 036440 .ASCII / 4 = BOUNCE BETWEEN I. D. & O. D. /<CR><LF>
6829 034145      040 020040 020065 .ASCII / 5 = BOUNCE BETWEEN INCR. O. D. & DECR. I. D. /<CR><LF>
6830 034224 020040 033040 036440 .ASCII / 6 = BOUNCE BETWEEN O. D. & DECR. I. D. /<CR><LF>
6831 034275      011 027117 027104 .ASCII / O. D. = OUTSIDE DIA. & I. D. = INSIDE DIA. /<CR><LF>
6832 034350 054524 042520 021040 .ASCIZ /TYPE "CR" TO CONTINUE/
6833 034376 054105 051105 044503 MSG8: .ASCIZ /EXERCISE # (0-6)/
6834 034417      104 052101 020101 MSG11: .ASCIZ /DATA PATTERN # (0-6)/
6835 034444 051124 041501 020113 MSG14: .ASCIZ /TRACK SEQUENCE # (0-6)/
6836 034473      111 020123 042524 MSG15: .ASCIZ /IS TEST TO RUN IN DOUBLE DENSITY /
6837 034535      111 020123 042524 MSG16: .ASCIZ /IS TEST TO RUN IN DELETED DATA MODE /
6838 034602 047101 020131 051120 MSG17: .ASCIZ /ANY PROGRAM CONTROL FLAGS /
6839 034635      040 020040 042522 MSG18: .ASCIZ / RETRY ON ERROR, LOG SOFT & HARD ERRORS /
6840 034710 020040 051040 041505 MSG19: .ASCIZ / RECALIBRATE ON SEEK ERRORS /
6841 034747      040 020040 051120 MSG20: .ASCIZ / PRINT ONLY 10 DATA ERRORS & CONTINUE /
6842 035020 020040 041440 042514 MSG21: .ASCIZ / CLEAR STATISTICAL TABLES BEFORE NEXT PASS /
6843 035076 047515 044504 054506 MSG22: .ASCIZ /MODIFY TRACK ADDRESS LIMITS /
6844 035133      040 020040 052517 MSG23: .ASCIZ / OUTER DIAMETER ADR # /
6845 035164 020040 044440 047116 MSG24: .ASCIZ / INNER DIAMETER ADR # /
6846 035215      115 042117 043111 MSG25: .ASCIZ /MODIFY SECTOR ADDRESS LIMITS /
6847 035253      040 020040 044515 MSG26: .ASCIZ / MIN. SECTOR ADR # /
6848 035301      040 020040 040515 MSG27: .ASCIZ / MAX. SECTOR ADR # /
6849 ;-----
6850      035330      . EVEN

```

6852
6853 035330 000232
6854
6855
6856 035562 000400
6857
6858
6859 036162 000400
6860
6861
6862
6863
6864 036562 000000
6865 037164
6866
6867
6874
6875
6876 037164
(3) 037164
6877 037164

```
-----  
TRKTBL: . REPT 154. ; TRACK TABLE  
         . BYTE 000  
         . ENDM  
DATPAT: . REPT 256. ; DATA PATTERN  
         . BYTE 000  
         . ENDM  
DATBUF: . REPT 256. ; DATA BUFFER  
         . BYTE 000  
         . ENDM  
-----  
PATCH: 0 ; PATCH AREA  
       . = +400  
-----  
  
LASTAD  
LSLAST: ENDMOD
```

6879
17750 067760 000000
17751 067762 000000
17752 067764 000000
17753 067766 000000
17754 067772
17755 000200

.SBTTL DIAGNOSTIC SUPERVISOR -- LOW CORE SET UP
.WORD 0 ;SPACE FOR USER POOL POINTER
.WORD 0 ;SIZE
.WORD 0 ;CHECKSUM (NOT CURRENTLY USED)
.WORD 0 ;SIZE OF H. W. PTAB. ALLOCATION
END. SUPV= +2
.END 200

EC3	030576	6578	6602#					
EC4	030633	6579	6603#					
EC5	030714	6580	6604#					
EC6	031005	6581	6605#					
EC7	031044	6582	6606#					
EDB241	022266	5771#						
EDC20	015622	4920	4984#					
ED1211	012050	4521	4523	4525	4527	4529#		
ED2341	021322	5623	5626#					
EF.COM=	000036 G	3213#	3251#	3929				
EF.MEM=	000035 G	3213#	3252#					
EF.PHR=	000034 G	3213#	3253#					
EF.RES=	000037 G	3213#	3250#	3935				
EF.STA=	000040 G	3213#	3249#					
EF01 =	000001 G	3213#	3270#					
EF02 =	000002 G	3213#	3269#					
EF03 =	000003 G	3213#	3268#					
EF04 =	000004 G	3213#	3267#					
EF05 =	000005 G	3213#	3266#					
EF06 =	000006 G	3213#	3265#					
EF07 =	000007 G	3213#	3264#					
EF08 =	000010 G	3213#	3263#					
EF09 =	000011 G	3213#	3262#					
EF10 =	000012 G	3213#	3261#					
EF11 =	000013 G	3213#	3260#					
EF12 =	000014 G	3213#	3259#					
EF13 =	000015 G	3213#	3258#					
EF14 =	000016 G	3213#	3257#					
EF15 =	000017 G	3213#	3256#					
EF16 =	000020 G	3213#	3255#					
EG1211	011652	4498	4500	4502#				
EIA11	007002	4001	4005	4015#				
EIA11	010342	4305	4307#					
EIA12	010624	4341	4351	4358#				
EIA121	011362	4407	4440	4442	4444#			
EIA23	016762	5218	5220#					
EIA24	021616	5695	5697#					
EIA25	025334	6253	6282#					
EIA40	027046	6454	6477#					
EIB11	010404	4313	4315#					
EIB121	011214	4416	4419#					
EIB20	015570	4957	4963	4975#				
EIB23	017060	5230	5240#					
EIB234	021230	5582	5588	5593	5598#			
EIB251	026246	6354	6363	6366	6369	6378	6381	6385#
EIB26	026600	6411	6413#					
EIC11	006774	4010	4014#					
EIC11	010432	4319	4321#					
EIC121	011336	4426	4430	4438#				
EIC20	015320	4930	4935#					
EIC25	025266	6265	6267	6273#				
EIC40	027026	6465	6467	6473#				
EID23	017234	5261	5268#					
EID232	020362	5473	5477#					
EID233	020570	5523	5525#					
EIE12	011034	4385	4389#					

ELE234	021202	5591	5594#					
ELE244	025010	6198	6201#					
ELF12	010772	4375	4382#					
ELF20	015346	4938	4940#					
ELF231	017776	5380	5386#					
ELF232	020300	5460	5463#					
ELF251	026234	6373	6382#					
ELG11	007044	4023	4027#					
ELG12	011076	4391	4394#					
ELG21	016144	5053	5057#					
ELG251	026134	6360	6364#					
ELH11	007110	4032	4036#					
ELH12	010642	4359	4362#					
ELH20	015436	4951	4954#					
ELH231	017646	5354	5359#					
ELH234	021036	5553	5572#					
ELH251	026222	6375	6379#					
ELI11	007300	4041	4069#					
ELJ21	016272	5072	5077#					
ELK11	007170	4046	4050#					
ELK20	015156	4903	4911#					
ELK234	020764	5559	5562#					
ELL11	007232	4054	4058#					
ELL20	015502	4959	4962#					
ELM242	023076	5860	5863#					
ELN21	016112	5047	5050#					
ELT242	023402	5906	5909#					
EHT. TR	037514	6980#	7776#	9993	9994	11637#		
ENDCVT	002524	3610	3612	3616	3619#			
END11	007350	4068	4070	4073#				
ENDLD	014146	4705	4710#					
ENDRPT	003036	3743#						
ENDST	006326	3903#	4887					
ENDTKS	015040	4745	4863	4867#				
ENDUP	030036	6539	6543#					
ENDXER	030354	6560	6570#					
END. OF	045540	9850	9943#					
END. SU	067772	8042#	8107	17754#				
END00	010236	4258	4267#					
END121	011410	4448	4450#					
END13	013600	4588	4592#					
END131	014106	4641	4659	4677	4692	4695#		
END133	015106	4892#						
END20	015636	4949	4981	4986#				
END22	016710	5200#						
END231	020036	5385	5388	5390	5391	5396#		
END232	020414	5463	5478	5483#				
END233	020606	5529#						
END234	021234	5549	5599#					
END241	022242	5758	5760	5762	5764#			
END242	023406	5805	5809	5827	5895	5908	5910#	
END243	024122	6026#						
END244	025024	6182	6190	6196	6200	6202	6204#	
END251	026270	6349	6371	6387#				
END26	026624	6405	6421#					
END30	026646	6428	6430#					

MM. ADR	037220 G	6935#	8033	8043	
MSRAB	063256	16646	16697#		
IA1211	011456	4471	4473#		
IA24U1	024574	6151#			
IA243	023560	5958#			
IA2431	024212	6070#			
IA2432	024256	6090#			
IA2433	024452	6125#			
IBU234	021346	5636#	5643		
IB1211	011466	4475#			
IB24U1	024604	6153#			
IB243	023646	5971#			
IB2432	024266	6092#			
IB2433	024470	6129#			
ICU234	021354	5638#			
IC1211	011512	4474	4480#		
IC243	023706	5982#			
IC2432	024354	6096	6103#		
ID	015060	4308#	4583#	4737	4876#
IDCOMP	014716	4844#			
IDENT1	030046	6537	6548#		
ID1211	011550	4485	4487#		
ID243	023622	5966#			
IEU234	021404	5646#			
IE1211	011570	4490#			
IFA11	006706	3997#			
IFAUP	027704	6538#			
IFAU23	021452	5663#			
IFAOO	010126	4242#			
IFA10	010256	4281#			
IFA11	010314	4302#			
IFA12	010460	4336#			
IFA121	011126	4406#			
IFA20	015162	4912#	4919		
IFA21	015734	5015	5018	5020#	
IFA22	016530	5172#			
IFA23	016726	5213#			
IFA231	017604	5348#			
IFA232	020224	5448#			
IFA233	020464	5505#			
IFA234	020660	5546#			
IFA24	021602	5694#			
IFA241	021702	5719#			
IFA242	022456	5795#			
IFA244	024712	6181#			
IFA25	025126	6242#			
IFA251	026042	6348#			
IFA26	026536	6404#			
IFAO	026674	6444#			
IFB11	006726	3998	4002#		
IFB00	010134	4244#			
IFB12	010540	4346#			
IFB121	011154	4412#			
IFB13	013546	4585#			
IFB20	015250	4925#			
IFB21	015752	5021	5024#		

IFB22	016540	5174#		
IFB23	017020	5222	5229#	
IFB231	017700	5366#		
IFB232	020320	5465	5467#	
IFB233	020516	5506	5511	5513#
IFB242	022554	5796	5810#	
IFB244	024730	6187#		
IFB25	025176	6258#		
IFB251	026050	6350#		
IFB26	026552	6408#		
IFB40	026740	6458#		
IFC11	006742	4003	4006#	
IFC00	010144	4243	4247#	
IFC11	010406	4316#		
IFC12	010650	4357	4361	4363#
IFC121	011262	4424	4427#	
IFC13	013554	4587#		
IFC20	015264	4928#		
IFC21	015774	5025	5029#	
IFC22	016566	5177	5179#	
IFC23	017126	5249	5251#	
IFC231	017722	5358	5361	5372#
IFC232	020336	5471#		
IFC233	020524	5515#		
IFC234	021016	5568#		
IFC241	022004	5736#		
IFC242	022514	5802#		
IFC244	024742	6188	6191#	
IFC25	025232	6259	6266#	
IFC251	026072	6351	6355#	
IFC40	026774	6459	6466#	
IFD11	006762	4007	4011#	
IFD00	010160	4251#		
IFD12	010660	4364	4366#	
IFD121	011302	4428	4431#	
IFD21	016026	5036#		
IFD22	016610	5173	5184#	
IFD23	017144	5255#		
IFD231	017740	5377#		
IFD232	020346	5472	5474#	
IFD233	020546	5514	5518	5520#
IFD234	021026	5570#		
IFD241	022014	5738#		
IFD242	022672	5825	5828#	
IFD244	024750	6193#		
IFD25	025316	6278#		
IFD251	026102	6357#		
IFE11	007004	4016#		
IFE00	010172	4254#		
IFE12	010670	4368#		
IFE121	011310	4433#		
IFE21	016344	5088#		
IFE22	016636	5187	5189#	
IFE23	017266	5273	5275	5277#
IFE232	020370	5475	5479#	
IFE233	020504	5510#		

IFE234	021162	5590#			
IFE241	022030	5737	5739	5741#	
IFE242	022706	5829	5831#		
IFE244	024770	6194	6197#		
IFE251	026160	6356	6370#		
IFF11	007012	4018#			
IFF12	010726	4374#			
IFF121	011320	4435#			
IFF20	015334	4937#			
IFF21	015726	5017#			
IFF23	017274	5279#			
IFF231	017746	5379#			
IFF232	020264	5459#			
IFF241	022040	5743#			
IFF242	023150	5869	5872#		
IFF251	026170	6372#			
IFG11	007020	4021#			
IFG12	011042	4365	4367	4381	4390#
IFG121	011350	4432	4434	4436	4441#
IFG20	015354	4941#			
IFG21	016122	5030	5032	5052#	
IFG23	017242	5252	5254	5270#	
IFG231	017616	5351#			
IFG232	020302	5449	5464#		
IFG242	022734	5812	5837#		
IFG251	026112	6359#			
IFH11	007062	4020	4030#		
IFH12	010630	4359#			
IFH121	011242	4423#			
IFH20	015422	4942	4950#		
IFH21	016160	5023	5060#		
IFH23	017356	5276	5290#		
IFH231	017626	5353#			
IFH232	020156	5436#			
IFH234	020706	5552#			
IFH242	022764	5838	5843#		
IFH251	026200	6374#			
IF111	007126	4017	4039#		
IF120	015276	4929	4931#		
IF121	016202	5061	5063	5065#	
IF123	016764	5221#			
IF1231	017542	5335#			
IF1241	021706	5721#			
IF1242	022744	5839#			
IFJ11	007136	4042#			
IFJ21	016236	5066	5071#		
IFJ23	017120	5248#			
IFJ241	022206	5757#			
IFJ242	022774	5840	5845#		
IFK11	007146	4045#			
IFK20	015112	4902#			
IFK21	016002	5031#			
IFK234	020744	5558#			
IFK241	022214	5759#			
IFK242	023022	5848	5851#		
IFL11	007206	4044	4053#		

IFL20	015460	4932	4958#						
IFL21	015714	5014#							
IFL241	022224	5761#							
IFL242	023104	5842	5850	5856	5862	5864#			
IFM20	015572	4976#							
IFM21	016062	5044#							
IFM241	022060	5747#							
IFM242	023050	5854	5857#						
IFN21	016072	5046#							
IFN242	023130	5834	5844	5865	5868#				
IF021	016304	5079#							
IF0242	023170	5877#							
IFP242	023176	5879#							
IFQ242	023216	5878	5883#						
IFR242	023234	5874	5876	5880	5882	5884	5886#		
IFS242	023276	5836	5871	5890	5892	5894#			
IFT242	023364	5905#							
IFU242	023252	5887	5889#						
IFV242	023260	5891#							
IFX242	022652	5824#							
IF1211	011610	4494#							
IG1211	011626	4495	4497#						
IG243	023740	5962	5990#						
IH1211	011700	4506#							
IH243	023774	5993	5999#						
I11211	011762	4516#							
I1243	024034	6002	6010#						
IJ1211	011772	4518#							
IJ243	024046	6013#							
INCSAD	032460	6627#	6637#	6649	6655#				
INCTRK	015654	4964#	4972#	4994#	5279	5281			
INDITK	002170	3131#	4308	4583	5434				
ININIT	037442 G	6964#	10088#	10775	11635#	11780			
INIT	006330	3929#							
INITER	006474	3934	3950#						
INITIA	056470	13712#							
INITL	010242	4232#	4270#	4902	4974#	5014	5229	5335	
INITTK	020454	4904#	5436	5438#	5497#				
INIT. M	044554	7831	8803	8827	9631#				
INIT. R	037256 G	6952#	10003	11638					
INMSG2	007366	4069	4082#						
INMSG3	007434	4071	4083#						
INPUTA	057416	7560	13360	14576#					
INTCMD	011422	4414#	4417#	4419	4457#				
INTER	004502	3885#							
INTER1	006622	3950	3986#						
INTER2	012116	4339	4387	4545#					
INTER3	012214	4349	4383	4546#					
INTER4	012314	4392	4547#						
INTFOR	052714	12302	12378#						
INTHD	032374	3953	6626#						
INTH1	032412	3957	6636#						
INTLV	020104	5351	5359	5369	5413#				
INVAL.	043522	7592	8817	8994#					
INVINT	052552	6896	12299#						
INV. SW	040554	7576	7671#						

LSTIML	002014	G	3061#							
LSTIMU	002054	G	3061#							
LSTIM1	002052	G	3061#							
LSTST1	002100	G	3061#							
LSUNIT	002012	G	3061#	3933	3944					
L. CLK.	043562		9001#	9017						
L10000	002150		3099	3112#						
L10001	002176		3122	3142#						
L10002	002330		3493#							
L10003	003040		3744#							
L10004	006672		3969	3989#						
L10005	007520		4104	4120#						
L10006	010030		4159	4182#						
L10007	010036		4201	4217#						
L10010	010240		4268#							
L10011	032532		6721	6728	6735#					
L10012	033052		6763	6788	6799#					
MAJ. IN	037246	G	6948#	11678#	11695#	17173#	17259	17291		
MAJ. LO	064656		17137#	17210	17259#	17283#	17290#	17291#	17292	
MAJ. US	037250	G	6949#	11889	17292#					
MAN. TI	001244		8097	8129#						
MAP16	067364	G	9481	9540	9592	17633#				
MASK. B	046266		9480	9485	9490	9539	9544	9591	9596	10365#
MASK. H	046264		9479	9482	9538	9541	9590	9593	10363#	
MAXSEC	002174		3133#	5366	5372	5377	5379			
MAXTRK	020436		5434#	5464	5471	5474	5479	5490#		
MCRDRT	026503		6380	6395#						
MCHTRT	026452		6365	6394#						
MDRDRT	026375		6377	6392#						
MHWTRT	026316		6362	6390#						
MEM. SI	043624		9013#	9024						
MINSEC	002172		3132#	5348	5362	5363	5364	5366		
MINTRK	020434		5435#	5444	5466	5467	5479	5482	5489#	
MIN. IN	037242	G	6946#	11679#	11696#	17174#	17260	17263	17288	
MIN. US	037244	G	6947#	11891	17289#					
MOOR	066726	G	8428	17424#						
MORDT	026426		6383	6393#						
MSG. AD	037210	G	6931#							
MSG. TY	037164	G	6923#	8169#	8172#	8982#				
MSG1	032532		6723	6737#						
MSG11	034417		6769	6834#						
MSG14	034444		6770	6835#						
MSG15	034473		6771	6836#						
MSG16	034535		6772	6837#						
MSG17	034602		6773	6838#						
MSG18	034635		6775	6839#						
MSG19	034710		6776	6840#						
MSG2	032545		6724	6738#						
MSG20	034747		6777	6841#						
MSG21	035020		6778	6842#						
MSG22	035076		6779	6843#						
MSG23	035133		6781	6844#						
MSG24	035164		6782	6845#						
MSG25	035215		6783	6846#						
MSG26	035253		6785	6847#						
MSG27	035301		6786	6848#						

MSG3	032564	6725	6739#									
MSG4	032574	6726	6740#									
MSG5	033052	6787	6805#									
MSG6	033106	6765	6806#									
MSG7	033127	6767	6807#									
MSG8	034376	6768	6833#									
MSKRT	026272	6353	6389#									
MUL	066662 G	8039	9402	9820	17407#							
MWTRT	026350	6368	6391#									
NEWPR1	066406 G	8005	11687	17177	17209	17329#						
NEWTRK	014776	4772	4782	4805	4822	4837	4856	4858#				
NEXTAR	062646	16478#	16890	16971	17031							
NOERL	004462	3883#										
NO. CLK	043552	8998#										
NO. FLA	062442	16263	16285#									
NO. LPT	057760	14713	14737#									
NO. PTA	043756	8754	9034#									
NR =	000001	3310#										
NSEC	020072	5345	5408#									
NUMBIN	053104	12603#	13397	13428	15578	15590	15842	15941	16927	16950	17019	
NUM. LA	053252	12636	12641#									
NUM. NO	037226 G	6938#	8092									
NUM. UN	037634	7356#	7612	8411	8969							
NUMITS	046242	9805	10358#									
NXSCSA	020054	5397#	5401#									
NXTFOR	062722	16537#	16894	16898	16906							
OCTMSG	056446	13228	13620#									
OD	015056	4307#	4582#	4735	4739	4875#						
ODCOMP	014730	4845	4847#									
ONEFIL =	000001	1898#	3039	3145	3146	3159	3647	3648	3661	6676		
OTDITK	002166	3130#	4307	4582	5226	5237	5435					
OTDVM	020612	5266#	5302#	5537#								
OTERTP	025106	4978#	6237#									
OTEXCH	026626	4261#	6426#									
OTSYER	026650	4253#	6435#									
OUTSID	021260	5544	5557	5565	5576	5581	5587	5597	5618#			
OSAPTS =	000001	3018#	3051#	3061								
OSAU =	000001	3018#	3051#	3061								
OSBGR =	000001	3018#	3051#	3061								
OSBNS =	000001	3018#	3051#	3061								
OSDU =	000001	3018#	3051#	3061								
OSGSM =	000001	3018#	3051#	3061								
OSPOIN =	000001	3018#	3051#	3061								
OSPIR =	000001	3051#										
O. OOT =	888888 U	6912										
PAR	003330	3760#	3777#	3785#	3807	3810#						
PARSES	062156	7566	16146#									
PAR. LA	056144	13215	13512#									
PASS. C	037200 G	6929#	8417#									
PAT	014154	4577#	4615	4617#	4618	4714#						
PATCH	036562	6864#										
PAT125	014026	4632	4674#									
PAT333	014052	4633	4683#									
PG	013730	4639#	4642	4647								
PLOC	006616	3946#	3983#	3996								
PREPT1	003302	3762#	3779#	3789#	3806#							

TRKTBL	035330	4858	5453	6853#										
TSAVCT	016370	5069#	5075#	5077#	5079	5098#								
TSEC	002317	3397#	6541											
TST	015640	4924#	4925	4928	4937	4965	4970	4988#						
TSTCK	024146	5697#	6037#	6090	6103	6129								
TSTEV	021634	4968#	5689	5692	5694	5697	5702#	5845	5851	5857	5872	5924		
TSTM	002156	3126#	4922											
TSTPAT	002160	3127#	4577											
TSTPTR	016360	5016#	5024	5026#	5033#	5048#	5050#	5055#	5058#	5067#	5073#	5078#	5085	5090#
		5094#												
TSTSUT	016712	5176#	5178#	5179	5186#	5188#	5189	5202#						
TSTM0	016364	4924	5036	5044	5046	5087#	5096#							
TST.AB	046400	10774#	11488	12124										
TST.TO	040602	7620	7675#											
TTRK	002316	3396#	6541											
TYPEC	057006	13944	14166	14242#	14589	14598	14623	14630						
TYPEPC	052702	12352#												
TYPFLA	062324	8859	16256#											
TYPLIN	056704	7621	8001	8749	9834	9851	9910	11597	12210	12638	12663	13409	13458	13506
		14079#	14714	16180	16258	17736								
TYPNUM	056266	8854	8940	9855	9914	13328	13568#							
TYPSTR	056724	8947	13329	13591	14082	14155#	14581	16647	17088					
TYP.ER	052532	12135	12244#											
TY.UNI	045544	9909	9945#											
TSARGC=	000001	3061#	3807#	3820#	3950#	4069#	4071#	4142#	4151#	4392#	4539#	5750#	5751#	6385#
		6537#	6540#	6541#	6568#									
TSCODE=	001004	4489#	6723#	6724#	6725#	6726#	6728#	6765#	6766#	6767#	6768#	6769#	6770#	6771#
		6772#	6773#	6774#	6775#	6776#	6777#	6778#	6779#	6780#	6781#	6782#	6783#	6784#
		6785#	6786#	6787#	6788#									
TSERCO=	000021	6264#	6271#	6464#	6472#									
TSERN=	000000	3018#	6264#	6271#	6464#	6472#								
TSEXCP=	000000	6723#	6724#	6725#	6726#	6768#	6769#	6770#	6781#	6782#	6785#	6786#		
TSFLAG=	000041	3969#	4104#	4159#	4201#	6728#	6788#							
TSHILI=	000032	6723#	6724#	6725#	6726#	6768#	6769#	6770#	6781#	6782#	6785#	6786#		
TSLOLI=	000001	6723#	6724#	6725#	6726#	6768#	6769#	6770#	6781#	6782#	6785#	6786#		
TSLSYM=	010000	3018#	3112	3142	3493	3744	3989	4120	4182	4217	4268	6735	6799	
TSNEST=	177777	3018#	3044#	3099#	3112#	3122#	3142#	3144#	3206#	3466#	3493#	3645#	3698#	3706#
		3744#	3927#	3989#	4095#	4120#	4130#	4182#	4193#	4217#	4225#	4249#	4260#	4268#
		4921#	4982#	6721#	6728	6735#	6763#	6766	6774	6780	6784	6788	6799#	6877#
TSNSKO=	000000	3044#	3144	3206#	3645	3698#	6877							
TSNSK1=	000005	3099#	3112	3122#	3142	3466#	3493	3706#	3744	3927#	3989	4095#	4120	4130#
		4182	4193#	4217	4225#	4268	4921#	4982	6721#	6728	6735	6763#	6766	6774
		6780	6784	6788	6799									
TSNSK2=	000003	4249#	4260											
TSSAVL=	177777	3018#												
TSSEGL=	177777	3018#	4249#	4260#	4921#	4982#								
TSSEKO=	010001	4249#	4260	4921#	4982									
TSSUBN=	000000	3018#	4225#											
TSTAGL=	177777	3018#												
TSTAGN=	010013	3018#	3099#	3122#	3466#	3706#	3927#	4095#	4130#	4193#	4225#	6721#	6763#	
TSTEMP=	000000	3081#	3112#	3142#	3144#	3493#	3645#	3744#	3969#	3989#	4104#	4120#	4159#	4182#
		4201#	4217#	4260#	4268#	4489#	4982#	6723#	6724#	6725#	6726#	6728#	6735#	6765#
		6767#	6768#	6769#	6770#	6771#	6772#	6773#	6775#	6776#	6777#	6778#	6779#	6781#
		6782#	6783#	6785#	6786#	6787#	6788#	6799#	6877#					
TSTEST=	000001	3018#	4225#											
TSTSTM=	177777	3018#	3493	3744	3807	3820	3929	3935	3946	3950	3952	3953	3957	3969

9947#	10267#	10364#	10366#	11938#	12385#	12388#	12391#	12393#	12677#	13527#	13531#	14739#
14818#	16287#	16698#	17748#	17754								

ABORTW	138	30188					
BCOMPL	178	30188	3930				
BERROR	218	30188					
BGNALJ	258	30188	4193				
BGNCLN	338	30188	4095				
BGNDOJ	418	30188	4130				
BGNHRD	498	30188	6721				
BGNM4	608	30188	3099				
BGNINI	718	30188	3927				
BGNMOD	798	30188	3044	3206	3698		
BGNMSG	928	30188	3466				
BGNRPT	1008	30188	3706				
BGNSEG	1088	30188	4249	4921			
BGNSFT	1178	30188	6763				
BGNSRV	1288	30188					
BGNSUB	1368	30188					
BGNSW	1608	30188	3122				
BGNTST	1718	30188	4225				
BNCOMP	2018	30188	3936	3947	4488	5018	5249
BNERRO	2058	30188					
BREAK	2098	30188					
BRESET	2138	30188	4332				
BUFFER	2178	30188					
CKLOOP	2238	30188	4259				
CLKOFF	2278	30188					
CLKON	2318	30188					
CLREF	2408	30188					
CLRVEC	2458	30188					
COMMEN	2508	30188					
DEVREG	2668	30188	3310				
DEVTYP	2818	30188	3429				
DISPAT	2868	30188	3081				
DOCLN	3008	30188	3952	4257			
DODU	3048	30188	4564				
DORPT	3098	30188	4266				
ENDALJ	3138	30188	4217				
ENDCLN	3258	30188	4120				
ENDCOM	3378	30188					
ENDDU	3538	30188	4182				
ENDHRD	3658	30188	6735				
ENDM4	3778	30188	3112				
ENDINI	3878	30188	3989				
ENDMOD	3998	30188	3144	3645	6877		
ENDMSG	4128	30188	3493				
ENORPT	4248	30188	3744				
ENDSEG	4368	30188	4260	4982			
ENDSFT	4508	30188	6799				
ENDSRV	4628	30188					
ENDSUB	4788	30188					
ENDSW	4968	30188	3142				
ENDTST	5068	30188	4268				
EQUALS	5248	30188	32138				
ERRDF	5988	30188	6464				
ERRHRD	6028	30188	6271				
ERRSF	6068	30188	6472				
ERRSOF	6108	30188	6264				

ESCAPE	614#	3018#														
EXIT	641#	3018#	3969	4104	4159	4201	6728	6788								
FEQUAL	684#	3018#														
GETPRI	700#	3018#														
GETTIM	705#	3018#														
GMANIA	710#	3018#														
GMANID	719#	3018#														
GMANIL	728#	3018#	4489													
GPHARD	737#	3018#	3946													
GPRMA	743#	3018#	6723	6724												
GPRMD	766#	3018#	6725	6726	6768	6769	6770	6781	6782	6785	6786					
GPRML	788#	3018#	4489#	6765	6767	6771	6772	6773	6775	6776	6777	6778	6779	6783	6787	
HEADER	805#	3018#	3061													
INLOOP	813#	3018#	5017	5248												
ISETU	817#	3018#														
IOSTAR	825#	3018#														
LASTAD	833#	3018#	6876													
MANUAL	839#	3018#	4487													
MSBYTE	1632#	3018#	3061#													
MSCHEC	1830#	3018#	3969#	4104#	4159#	4201#	6728#	6788#								
MSCKID	1658#	3018#	3061#													
MSCOUN	1770#	3018#	3807#	3820#	3950#	4069#	4071#	4142#	4151#	4392#	4539#	5750#	5751#	6385#	6537#	
	6540#	6541#	6568#													
MSDATA	1519#	3018#	3061#	3429#												
MSDECR	1733#	3018#	3112#	3142#	3144#	3493#	3645#	3744#	3989#	4120#	4182#	4217#	4260#	4268#	4982#	
	6735#	6799#	6877#													
MSDEFA	1882#	3018#	4489#	6723#	6724#	6725#	6726#	6765#	6767#	6768#	6769#	6770#	6771#	6772#	6773#	
	6775#	6776#	6777#	6778#	6779#	6781#	6782#	6783#	6785#	6786#	6787#					
MSENDE	1778#	3018#	3112#	3142#	3144#	3493#	3645#	3744#	3989#	4120#	4182#	4217#	4260#	4268#	4982#	
	6735#	6799#	6877#													
MSERRI	1319#	3018#	6264#	6271#	6464#	6472#										
MSESCA	1638#	3018#														
MSESCS	1642#	3018#														
MSEXCP	1813#	3018#	6723#	6724#	6725#	6726#	6768#	6769#	6770#	6781#	6782#	6785#	6786#			
MSEXIT	1646#	3018#	3969#	4104#	4159#	4201#	6728#	6788#								
MSEXSE	1654#	3018#	3969#	4104#	4159#	4201#	6728#	6788#								
MSEXTJ	1650#	3018#	3969#	4104#	4159#	4201#	6728#	6788#								
MSGEN	1742#	3018#	3061#	3081#	3099#	3112#	3122#	3142#	3310#	3429#	3466#	3493#	3706#	3744#	3927#	
	3989#	4095#	4120#	4130#	4182#	4193#	4217#	4225#	4260#	4268#	4489#	4982#	6721#	6735#	6763#	
	6799#	6876#														
MSGENB	1580#	3018#	4489#													
MSGETS	1739#	3018#	3112#	3142#	3144#	3493#	3645#	3744#	3989#	4120#	4182#	4217#	4260#	4268#	4982#	
	6728#	6735#	6766#	6774#	6780#	6784#	6788#	6799#	6877#							
MSGETT	1524#	3018#	3969#	4104#	4159#	4201#	6728#	6766#	6774#	6780#	6784#	6788#				
MSGNGB	1549#	3018#	3044#	3061#	3081#	3099#	3122#	3206#	3310#	3429#	3466#	3698#	3706#	3927#	4095#	
	4130#	4193#	6721#	6763#	6876#											
MSGNIN	1753#	3018#	3061#	3081#	3099#	3122#	3310#	3429#	3493#	3744#	3807#	3820#	3929#	3930#	3935#	
	3936#	3946#	3947#	3950#	3952#	3953#	3957#	3969#	3989#	4069#	4071#	4104#	4120#	4142#	4151#	
	4159#	4182#	4201#	4217#	4249#	4257#	4259#	4260#	4266#	4268#	4332#	4392#	4487#	4488#	4489#	
	4539#	4564#	4921#	4982#	5017#	5018#	5248#	5249#	5633#	5651#	5750#	5751#	6264#	6271#	6385#	
	6464#	6472#	6537#	6540#	6541#	6568#	6721#	6723#	6724#	6725#	6726#	6728#	6735#	6763#	6765#	
	6766#	6767#	6768#	6769#	6770#	6771#	6772#	6773#	6774#	6775#	6776#	6777#	6778#	6779#	6780#	
	6781#	6782#	6783#	6784#	6785#	6786#	6787#	6788#	6799#	6876#						
MSGNLS	1555#	3018#	4260#	4489#	4982#											
MSGNSU	1545#	3018#														
MSGNTA	1537#	3018#	3112#	3142#	3493#	3744#	3989#	4120#	4182#	4217#	4268#	6735#	6799#			

SETVEC	1080#	3018#	3953	3957					
SLASH	1086#	3018#							
STARS	1100#	3018#							
SVC	1114#	3017#	3018						
TRAPPR	1262#	3018#							
UNBUFF	1267#	3018#							
WAITMS	1272#	3018#							
WAITUS	1277#	3018#							
XFER	1282#	3018#	3969#	4104#	4159#	4201#	6728#	6788#	
XFERF	1286#	3018#	6766	6774	6780	6784			
XFERT	1290#	3018#							

ABS. 067770 000

ERRORS DETECTED: 0

CZRADA. BIN, CZRADA. SEQ/NL: BEX/CRF=CZRADA. SML, CZRADA. P11, DOCTOR. P11
RUN-TIME: 29 34 4 SECONDS
RUN-TIME RATIO: 295/68=4.2
CORE USED: 20K (39 PAGES)