

RX11, RX01
RX02

RX02 SS PERF EXER
CZRXDCO

AH-E513C-MC
FICHE 1 OF 1

SEP 1982
COPYRIGHT © 79-82
MADE IN USA



.REM 8

IDENTIFICATION

PRODUCT CODE: AC-E512C-MC
PRODUCT NAME: CZRxdc RX02 SS PERF EXER
PRODUCT DATE: 29-MAR-82
MAINTAINER: S.S.S.T.A.
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) 1979,1982 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
1.6	MEMORY MAP
2.0	OPERATING INSTRUCTIONS
2.1	HARDWARE QUESTIONS
2.2	SOFTWARE QUESTIONS
2.3	TIMING CONSIDERATIONS
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
3.7	ERROR NUMBERS
4.0	PERFORMANCE AND PROGRESS REPORTS
5.0	DEVICE INFORMATION TABLES
5.1	DEVICE PROTOCOL
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
7.0	LISTING INDEX
7.1	LISTING

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+ SUPERVISOR/USERS MANUAL CHQUS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

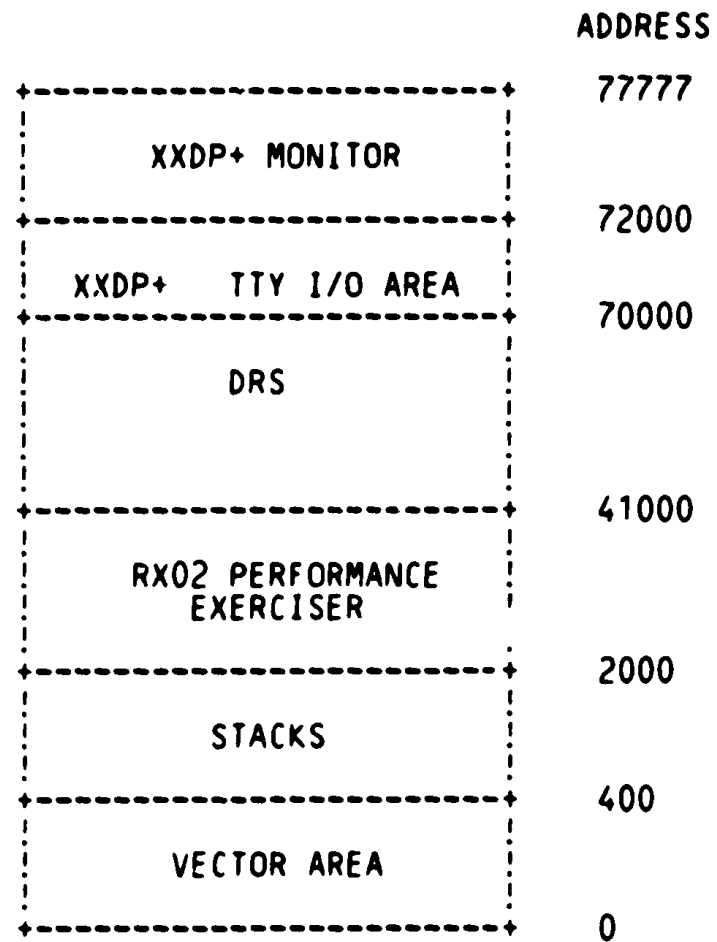
NONE

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.

1.6 MEMORY MAP

MEMORY LAYOUT ON 16K MACHINE - XXDP ENVIRONMENT



IN A MACHINE WITH MORE MEMORY FREE SPACE WILL OCCUR BETWEEN
THE DIAGNOSTIC AND THE DRS.

2.0 OPERATING INSTRUCTIONS

THIS IS A REV C SUPERVISOR DIAGNOSTIC: FOR OPERATING INSTRUCTIONS, PLEASE SEE CHAPTER 5 OF XXDP+ OPERATOR'S MANUAL. THEY ARE NO LONGER INCLUDED IN THE DIAGNOSTIC LISTING BECAUSE IT IS DESIRED THAT A CHANGE IN THOSE INSTRUCTIONS NOT REQUIRE A RE-ASSEMBLY OF ALL SUPERVISOR DIAGNOSTICS.

2.1 HARDWARE QUESTIONS

THE FOLLOWING SERIES OF QUESTIONS COMPRISE THE PARAMETERS NECESSARY TO IDENTIFY EACH FLOPPY DISK SUBSYSTEM.

RX BUS ADR -
THIS PARAMETER DEFINES THE BASE BUS ADDRESS FOR THE FLOPPY DISK SUBSYSTEM.

VECTOR ADR -
THIS PARAMETER DEFINES THE INTERRUPT VECTOR ADDRESS FOR THE FLOPPY DISK SUBSYSTEM INTERFACE.

DRIVE # -
THIS PARAMETER DEFINES THE FLOPPY DISK SUBSYSTEM DRIVE NUMBER (0 - 1).

EXP WRD-TYPE -
THIS PARAMETER IS TO BE USED FOR FUTURE EXPANSION. TYPE A CARRIAGE RETURN.

2.2 SOFTWARE QUESTIONS

EXERCISE # - ENTER # FROM TABLE SHOWN IN PARAGRAPH 6.3.

DATA PATTERN # - ENTER # FROM TABLE SHOWN IN PARAGRAPH 6.2.

TRACK SEQUENCE # - ENTER # FROM TABLE SHOWN IN PARAGRAPH 6.4.

DEVICE FATAL THRESHOLD LEVEL -
THE DEVICE FATAL THRESHOLD LEVEL (DFTL) IS INITIALLY SET=1.
THIS THRESHOLD LEVEL EQUALS THE # OF HARD ERRORS THAT
WILL CAUSE A DEVICE FATAL ERROR WHEN THE DRS 'EVL' FLAG
IS SET. THE 'EVL' FLAG WILL ALSO CAUSE 10 SOFT ERRORS
TO BE RECLASSIFIED A HARD ERROR, WHICH IF DFTL = 1 WILL
BECOME A DEVICE FATAL ERROR.

RUN TEST IN DOUBLE DENSITY -
IF TEST IS IN WRONG DENSITY - OPERATOR WILL BE ASKED IF
THE DISKETTE IS TO BE REFORMATTED.

RUN TEST IN DELETED DATA MODE -
IF ANSWERED YES, DELETED DATA MODE WILL BE DONE FIRST.

ANY PROGRAM CONTROL FLAGS -
IF ANSWERS YES THE FOLLOWING QUESTIONS WILL BE ASKED.
RETRY ON ERROR, LOG SOFT + HARD ERRORS?
IF RETRY IS NOT SET, THEN SOFT ERRORS
WILL ALSO LOG AS HARD ERRORS.
RECALIBRATE ON SEEK ERRORS?
PRINT ONLY 10 DATA ERRORS + CONTINUE?
CLEAR STATISTICAL TABLES BEFORE NEXT PASS?

MODIFY TRACK ADDRESS LIMITS -
IF ANSWERING YES, THEN THE FOLLOWING WILL BE ASKED:
OUTER DIAMETER ADR #?
INNER DIAMETER ADR #?

MODIFY SECTOR ADDRESS LIMITS -
IF ANSWERING YES, THEN THE FOLLOWING WILL BE ASKED:
MIN. SECTOR ADR #?
MAX. SECTOR ADR #?

RXXX EXPANSION <CR>
THIS WORD IS FOR FUTURE EXPANSION, ANSWER WITH A
CARRIAGE RETURN.

2.3 TIMING CONSIDERATIONS

TEST EFFICIENCY CAN BE IMPROVED WHEN RUNNING ON A LSI
PROCESSOR AS FOLLOWS:
11/03 CHANGE LOC 23706 FROM 3 TO 7 SAVES 33 MIN/PASS
11/23 CHANGE LOC 23706 FROM 3 TO 5 SAVES 30 MIN/PASS
THESE PATCHES OPTIMIZE THE INTERLEAVE FACTOR

3.0 ERROR INFORMATION

THIS PROGRAM HAS FOUR TYPES OF ERROR CLASSIFICATIONS; SYSTEM FATAL, DEVICE FATAL, HARD AND SOFT.

SYSTEM FATAL ERRORS

SYSTEM FATAL ERRORS ARE USED TO INDICATE THAT AN ERROR WAS DETECTED BY THE DIAGNOSTIC SUPERVISOR IN RELATION TO LOADING/ CONTROLLING THE DIAGNOSTIC PROCESS.

THE CONTENT OF EACH ERROR IS SUCH THAT IT SHOULD BE SELF - EXPLANATORY. HOWEVER, THE MESSAGES UTILIZE SOME TERMS THAT ARE SPECIFIC TO THE FLOPPY DISK SUBSYSTEM, AND MAY REQUIRE SOME GETTING USE TO.

DEVICE FATAL ERRORS

DEVICE FATAL ERRORS ARE A RESULT OF:

1. REACHING A DEVICE FATAL THRESHOLD LEVEL ('DFTL'). AN 'DFTL'=1 WILL CAUSE 1 HARD ERROR TO BE CLASSIFIED A DEVICE FATAL ERROR. THIS LEVEL IS INITIALLY SET=1, BUT MAY BE MODIFIED BY THE OPERATOR.
2. AN ERROR THAT IS CONSIDERED FATAL TO THE DEVICE, BUT TESTING WILL CONTINUE.

HARD ERRORS

HARD ERRORS ARE A RESULT OF:

1. TEN RETRIES OF A SOFT ERROR OR
2. A NON-RECOVERABLE ERROR

SOFT ERRORS

SOFT ERRORS ARE MEDIA RELATED ERRORS AND IF RETRY ON ERROR IS SET WILL BE TRIED UP TO TEN TIMES THEN CLASSIFIED AS HARD ERRORS.

IF RETRY ON ERROR IS NOT SET THE ERROR WILL BE LOGGED AS BOTH SOFT AND HARD ERRORS.

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>
2 - 253 (125) CONTAIN 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 RXV211 R*21/RX02 SUBSYSTEM.

3. ERROR NUMBERS

ERROR	- TYPE	- ERR #
SEEK	- SOFT	- 0 -32
CRC	- SOFT	- 1 -33
CKSUM	- HARD	- -34
DATA	- SOFT	- 3 -35
DEL. DATA UNEX	- HARD	- -37
DEL. DATA MISSING	- HARD	- -38
UNK ERR	- HARD	- -40
FILL/EMPTY BUFFER	- HARD	- -41
READ	- SOFT	- 10-42
WRITE	- SOFT	- 11-43
INTER-BUT NO DONE	- HARD	- -44
DONE-BUT NO INTER	- HARD	- -45
ERR-BUT NO ERR BIT	- HARD	- -46
ERR BIT SET	- HARD	- -47

NO DONE ON INIT	- SYS FATAL	- 128
NO DONE ON FUNCTION	- DEV FATAL	- 65
NO DRIVE RDY	- DEV FATAL	- 66
NO SIDE RDY	- DEV FATAL	- 67
NO DONE AFTER RD STA	- DEV FATAL	- 68
WRG DRV RESPOND	- SYS FATAL	- 133
WRG SIDE RESPOND	- SYS FATAL	- 134
DISKETT WRG DEN	- DEV FATAL	- 73
DENSITY ERR	- DEV FATAL	- 74
T.O. ON "TR" OR "DONE"	- SYS FATAL	- 139
SYS ERR	- SYS FATAL	- 140
INITIALIZE ERROR	- DEV FATAL	- 200
ADDRESSING ERROR	- SYS FATAL	- 400

- NOTES: 1. SOFT ERRORS HAVE TWO ERROR NUMBERS:
LOW # = SOFT ERROR
HIGH # = HARD ERROR (RECLASSIFIED SOFT ERROR)
2. IF "EVL" FLAG IS SET HARD ERRORS WILL BE RECLASSIFIED DEVICE FATAL ERRORS, BUT THE ERROR NUMBER WILL REFLECT THE ORIGINAL HARD ERROR.

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	0							TRACK ADDRESS
RXSA:	X	X	X	X	X	X	X	X	0	0	0					SECTOR ADDRESS
RXDB:	DATA BUFFER															

READ ERROR CODE REGISTERS - (SEE LABEL 'XERUUT')

WORD	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
#1	WORD COUNT								ERROR CODE							
#2	CURRENT TRACK DRV #1								CURRENT TRACK DRIVE #0							
#3	TARGET SECTOR								TARGET TRACK							
#4	BAD TRACK-ONLY VALID IF ERRCODE=150								UNT	DV1	HD	DVO	X	X	X	LCD
								SEL	DEN	LD	DEN					DEN

5.2 DEVICE PROTOCOL

RX02 FUNCTIONAL PROCESS

FUNCTION CODE BIT # 3 2 1	FUNCTION	PROCEDURE (PROTOCOL)
0 0 0	FILL BUFFER	FUNCTION WORD --->TR--->WC--->TR--->BA--->DONE
0 0 1	EMPTY BUFFER	FUNCTION WORD --->TR--->WC--->TR--->BA--->DONE
0 1 0	WRITE SECTOR	FUNCTION WORD --->TR--->SA--->TR--->TA--->DONE
0 1 1	READ SECTOR	FUNCTION WORD --->TR--->SA--->TR--->TA--->DONE
1 0 0	SET DENSITY	FUNCTION WORD --->TR--->VW--->DONE
1 0 1	READ MAINT. STATUS	FUNCTION WORD --->DONE
1 1 0	WRITE SECTOR WITH DELETED DATA	FUNCTION WORD --->TR--->SA--->TR--->TA--->DONE
1 1 1	READ ERROR CODE	FUNCTION WORD --->TR--->BA--->DONE

TR = WAIT FOR TR BIT
 DONE = WAIT FOR DONE BIT
 BA = BUS ADDRESS (OUTPUT TO RX)
 VW = VERIFICATION WORD (OUTPUT TO RX)
 WC = WORD COUNT (OUTPUT TO RX)
 SA = SECTOR ADDRESS (OUTPUT TO RX)
 TA = TRACK ADDRESS (OUTPUT TO RX)

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

17-	768	PROGRAM HEADER
17-	837	DISPATCH TABLE
19-	854	DEFAULT HARDWARE P-TABLE
19-	880	SOFTWARE P-TABLE
20-	924	GLOBAL EQUATES SECTION
22-	1076	GLOBAL DATA SECTION
26-	1194	GLOBAL TEXT SECTION
28-	1233	GLOBAL ERROR REPORT SECTION
28-	1241	- MOD U.SFT.ERR - ERROR REPORT
28-	1251	- MOD U.PRT.ERR - PRINT ERRORS
30-	1274	- MOD U.PRT.EC - PRINT UNIT ERROR CODE
32-	1342	- ERROR PRINT CALLS/MSG CALLS
34-	1375	GLOBAL SUBROUTINES SECTION
34-	1454	- MOD U.1.0 - RANDOM GENERATOR
36-	1480	- MOD U.A.1 - CONVERSION UUT CODE --> SUTPTR
36-	1504	- MOD U.A.2 - CONVERSION SUTPTR --> UUT CODE
38-	1525	- MOD U.DEV.REC - DEVICE READ ERROR CODE
39-	1564	REPORT CODING SECTION
41-	1653	- PRINT REPORT HEADER
41-	1674	- PRINT REPORT DATA
43-	1707	- PRINT READ/WRITE SECTOR COUNTERS
45-	1738	- PRINT REPORT TYPE 1
45-	1750	- PRINT REPORT TYPE 2
45-	1760	- PRINT REPORT TYPE 3
49-	1827	- STATISTICAL TABLES
49-	1870	LOAD DEVICE PROTECTION
51-	1881	INITIALIZE SECTION
53-	1958	- MOD I.1 - UNPACK HARDWARE P-TABLES
55-	2047	CLEANUP CODING SECTION
57-	2084	AUTO DROP SECTION
59-	2131	- TEST 0: ADDRESSING TEST
61-	2174	- MOD U.SFT.TRP - BUS TRAP HANDLER
63-	2194	DROP UNIT SECTION
65-	2251	ADD UNIT SECTION
67-	2285	TEST 1: RX02 SS PERF EXERCISER
67-	2289	MOD 0.0 - EXERCISE A SYSTEM
70-	2385	MOD 1.0 - GET SYSTEM EXERCISE
70-	2404	MOD 1.1 - GET EXERCISE CONDITIONS
72-	2433	MOD 1.2 - GET SYSTEM TO EXERCISE
72-	2500	- MOD 1.2.U.1 - GET PRINTABLE SYSTEM 0 UNIT #
72-	2517	- MOD 1.2.U.2 - GET PRINTABLE SYSTEM 1 UNIT #
74-	2531	MOD 1.2.1 - CK DRIVE AVAILABLE
78-	2603	MOD 1.2.1.1 - REFORMAT DRIVE DENSITY
80-	2685	- MOD 1.2.U.3 - INITIALIZE ERROR
80-	2698	- MOD 1.2.U.4 - INITIALIZE DROP
80-	2705	- MOD 1.2.U.5 - INITIALIZE PRINT
82-	2740	MOD 1.3 - GET EXERCISE

84-	2760	MOD 1.3.1 - SET DATA PATTERN
86-	2866	MOD 1.3.2 - SET TRACK SEQUENCE
86-	3015	MOD 1.3.3 - CLEAR STATISTICAL TABLES
88-	3029	MOD 2.0 - SCHEDULE SYSTEM EXERCISE
90-	3133	MOD 2.1 - GET A TEST
92-	3240	- EXERCISE/TEST TABLE
94-	3298	MOD 2.2 - GET A DRIVE
96-	3337	MOD 2.3 - EXECUTE DRIVE TEST
100-	3461	MOD 2.3.1 - GET A SECTOR
100-	3550	MOD 2.3.1.A - SET SECTOR DONE
102-	3562	MOD 2.3.2 - GET A TRACK
106-	3634	MOD 2.3.3 - GET A DRIVE FUNCTION
108-	3668	MOD 2.3.4 - OUTPUT DRIVE FUNCTION
108-	3743	MOD 2.3.4.1 - OUTPUT SINGLE WORD
110-	3757	MOD U.2.3.4 - WATCH DOG TIMER
110-	3787	MOD U.2.3/4 DELAY
112-	3815	MOD 2.4 - EVALUATE TEST RESULTS
114-	3833	MOD 2.4.1 - EVALUATE DATA
116-	3915	MOD 2.4.2 - EVALUATE DRIVE STATE
118-	4032	MOD 2.4.2.1 - EVALUATE DRIVE RESPONSE
120-	4065	MOD 2.4.3 - UPDATE DRIVE STATISTICS
122-	4178	MOD 2.4.3.1 - UPDATE HARD ERROR STATISTICS
122-	4193	MOD 2.4.3.2 - UPDATE CRC STATISTICS
124-	4222	MOD 2.4.3.3 - UPDATE SOFT ERROR STATISTICS
126-	4251	MOD 2.4.3.4 - UPDATE SECTOR WRITTEN/READ COUNTERS
128-	4284	- MOD 2.4.U.1 - SOFT ERROR LOGGER
130-	4317	MOD 2.4.4 - EVALUATE UNIT ERROR CODE
132-	4375	MOD 2.5 - OUTPUT ERROR TYPE
134-	4490	MOD 2.5.1 - PRINT RETRY
136-	4544	MOD 2.6 - SET DRIVES DONE
138-	4569	MOD 3.0 - OUTPUT EXERCISE COMPLETE
140-	4579	MOD 4.0 - OUTPUT SYSTEM ERROR
144-	4680	- MOD INTR.1 - INTERRUPT HANDLER #0
144-	4687	- MOD INTR.2 - INTERRUPT HANDLER #1
144-	4694	MOD U.INTR.U - SAVE UNIT REG
144-	4705	- READ ERROR CODE BUFFER
144-	4717	- TRACK TABLE
144-	4724	- DATA BUFFERS
146-	4748	HARDWARE PARAMETER CODING SECTION
148-	4824	SOFTWARE PARAMETER CODING SECTION
152-	4930	- PATCH AREA

7.1 LISTING

8

768
769

.DSABL GBL
.ENABL AMA,ABS

770
771
805
807
808
810
811 002000
812
813
814
815
816
817
818 002000
819
827
828 002000
829
835
836 002122
837
838
839
840
841
842
843
844
845
846
847 002152
848

.TITLE PROGRAM HEADER AND TABLES
.SBTTL PROGRAM HEADER

.ENABL ABS,AMA
= 2000

BGNMOD

:++
: THE PROGRAM HEADER IS THE INTERFACE BETWEEN
: THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
:--

POINTER BGNRPT,BGNSW,BGNSFT,BGNAU,BGNDU,ERRTBL,BGNSETUP

HEADER CZRXDC0,0,0,2100,1

: DESCRIPT ^\$RX02 SS PERF EXER \$
: .EVEN
:-----

.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

857
858
859
860
861
862
863
864
865 002156
866
867 002160 177170
868 002162 000264
869 002164 000000
870 002166 000000
871
877
878 002170
879
880
881
882
883
884
885
886
887
888
889
890 002170
891
892 002172 000000
893 002174 000000
894 002176 000000
895 002200 000000
896 002202 000000
897 002204 000021
898 002206 000000
299 002210 000114
900 002212 000001
901 002214 000032
902 002216 000001
903
910
911 002220
912
913 002220

.SBTTL DEFAULT HARDWARE P-TABLE

:++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
:--

BGNHW	DFPTBL	
.WORD	177170	:UNIBUS ADDRESS
.WORD	264	:VECTOR ADDRESS
.WORD	0	:DRIVE #
.WORD	0	:FUTURE EXPANSION

ENDHW

.SBTTL SOFTWARE P-TABLE

:++
: THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
: PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
:--

	BGNSW	SFPTBL	
RXXX:	.WORD	0	:FUTURE EXPANSION-RX
	.WORD	0	:P-TABLE CONTROL WORD
TSTN:	.WORD	0	:TEST #
TSTPAT:	.WORD	0	:TEST PATTERN #
TRKSEQ:	.WORD	0	:TRACK SEQUENCE #
SWREG:	.WORD	21	:SOFTWARE SWITCH REG
OTDITK:	.WORD	0	:OUTSIDE DIA. TRACK LIMIT
INDITK:	.WORD	114	:INSIDE DIA. TRACK LIMIT.
MINSEC:	.WORD	1	:MINIMUM SECTOR LIMIT
MAXSEC:	.WORD	32	:MAXIMUM SECTOR LIMIT
DFTL:	.WORD	1	:DEVICE FATAL THRESHOLD LVL

ENDSW

ENDMOD

(1)	000300	PRI06== 300
(1)	000240	PRI05== 240
(1)	000200	PRI04== 200
(1)	000140	PRI03== 140
(1)	000100	PRI02== 100
(1)	000040	PRI01== 40
(1)	000000	PRI00== 0
(1)		:
(1)		:OPERATOR FLAG BITS
(1)		:
(1)	000004	EVL== 4
(1)	000010	LOT== 10
(1)	000020	ADR== 20
(1)	000040	IDU== 40
(1)	000100	ISR== 100
(1)	000200	UAM== 200
(1)	000400	BOE== 400
(1)	001000	PNT== 1000
(1)	002000	PRI== 2000
(1)	004000	IXE== 4000
(1)	010000	IBE== 10000
(1)	020000	IER== 20000
(1)	040000	LOE== 40000
(1)	100000	HOE== 100000
983		:
984		:
985		:BIT DEFINITIONS
986		:
987	100000	BIT15== 100000
988	040000	BIT14== 40000
989	020000	BIT13== 20000
990	010000	BIT12== 10000
991	004000	BIT11== 4000
992	002000	BIT10== 2000
993	001000	BIT09== 1000
994	000400	BIT08== 400
995	000200	BIT07== 200
996	000100	BIT06== 100
997	000040	BIT15== 40
998	000020	BIT04== 20
999	000010	BIT03== 10
1000	000004	BIT02== 4
1001	000002	BIT01== 2
1002	000001	BIT00== 1
1003		:
1004	001000	BIT9== BIT09
1005	000400	BIT8== BIT08
1006	000200	BIT7== BIT07
1007	000100	BIT6== BIT06
1008	000040	BIT5== BIT05
1009	000020	BIT4== BIT04
1010	000010	BIT3== BIT03
1011	000004	BIT2== BIT02
1012	000002	BIT1== BIT01
1013	000001	BIT0== BIT00
1014		:

```
1015 ;EVENT FLAG DEFINITIONS
1016 ; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
1017 ; EF16:EF01 AVAILABLE FOR PROGRAM USE
1018 000040 EF.START== 32. ;START COMMAND WAS ISSUED.
1019 000037 EF.RESTART== 31. ;RESTART COMMAND WAS ISSUED.
1020 000036 EF.CONTINUE== 30. ;CONTINUE COMMAND WAS ISSUED.
1021 000035 EF.NEW== 29. ;A NEW PASS HAS BEEN STARTED.
1022 000034 EF.PWR== 28. ;A POWER FAIL/POWER-UP OCCURRED
1023 .
1024 000020 EF16== 16.
1025 000017 EF15== 15.
1026 000016 EF14== 14.
1027 000015 EF13== 13.
1028 000014 EF12== 12.
1029 000013 EF11== 11.
1030 000012 EF10== 10.
1031 000011 EF09== 9.
1032 000010 EF08== 8.
1033 000007 EF07== 7.
1034 000006 EF06== 6.
1035 000005 EF05== 5.
1036 000004 EF04== 4.
1037 000003 EF03== 3.
1038 000002 EF02== 2.
1039 000001 EF01== 1
1040 .
1041 ;PRIORITY LEVEL DEFINITIONS
1042 .
1043 000340 PRI07== 340
1044 000300 PRI06== 300
1045 000240 PRI05== 240
1046 000200 PRI04== 200
1047 000140 PRI03== 140
1048 000100 PRI02== 100
1049 000040 PRI01== 40
1050 000000 PRI00== 0
1051 .
1052 ;PROGRAM DEFINITIONS
1053 .
1054 000200 TRBIT==200
1055 000040 DNBIT==40
1056 004000 RX2BIT==BIT11
1057 000003 SOFT==3
1058 000002 HARD==2
1059 000001 DVFT==1
1060 000000 SYFT==0
1061 000004 BTRP4==4
1062 000006 BTRP6==6
1063 000002 RESTAR==BIT1
1064 000001 POWERF==BIT0
1065 004000 SYSERR==BIT11
```

```
1079          .SBTTL GLOBAL DATA SECTION
1080
1081          :++
1082          : THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1083          : IN MORE THAN ONE TEST.
1084          :--
1085
1086          :
1087          : STORAGE FOR DEVICE REGISTERS
1088          :
1089
1090          :-----
1091 002220 000000 UOADR: .WORD 0          :UNIT 0 ADR
1092 002222 000000 U1ADR: .WORD 0          :UNIT 1 ADR
1093 002224 000000 UOVECT: .WORD 0         :UNIT 0 VECTOR
1094 002226 000000 U1VECT: .WORD 0         :UNIT 1 VECTOR
1095          :-----
1096 002230 000000 SDD: .WORD 0          :SYSTEM DRIVES DONE (SEE REG. DEF. BELOW)
1097 002232 000000 SUT: .WORD 0          :SYSTEM UNDER TEST (SEE REG. DEF. BELOW)
1098 002234 000000 UUT: .WORD 0          :UNIT UNDER TEST (SEE REG. DEF. BELOW)
1099 002236 000000 UUTADR: .WORD 0         :UUT UNIBUS ADR
1100 002240 000000 UUTOFF: .WORD 0        :UUT TABLE ADDRESSING OFFSET
1101 002242 000000 DEN: .WORD 0          :DENSITY FLAG
1102 002244 000000 DELDAT: .WORD 0        :DELETED DATA FLAG
1103 002246 000000 CSRUUT: .WORD 0        :CONT/STATUS REG UUT
1104 002250 000000 ESRUUT: .WORD 0        :ERROR/STATUS REG UUT
1105          :-----
1106 002252 000000 WDCNT: .WORD 0         :WORD COUNT
1107 002254 000000 TRACK: .WORD 0        :TRACK ADR
1108 002256 000000 SECTOR: .WORD 0       :SECTOR ADR
1109 002260 000000 TRKDN: .WORD 0       :TRACK DONE (UUT) FLAG
1110 002262 000000 SECDN: .WORD 0       :SECTOR DONE (UUT) FLAG
1111          :-----
1112 002264 000000 FLGDRS: .WORD 0      :'DRS' FLAGS
1113 002266 000000 FLAGS: .WORD 0       :DIAGNOSTIC FLAGS
1114 002270 000000 ABORT: .WORD 0       :ABORT FLAG
1115 002272 000000 PRTECD: .WORD 0     :PRINT ERR CODE FLAG
1116          :-----
1117 002274 000000 ERRSY: .WORD 0       :ERROR SYSTEM
1118 002276 000000 ERRTY: .WORD 0       :ERROR TYPE
1119 002300 000000 HARDER: .WORD 0      :HARD ERROR
1120 002302 000000 HDERCT: .WORD 0     :HARD ERROR COUNTER (USED FOR 'DFTL')
1121          :-----
1122 002304 000000 RETRY: .WORD 0       :/(10)DATART/(4)RDRT/(2)WTRT/(?)SEEK/ SEE BELOW
1123 002306 000000 SEEKRT: .WORD 0      :SEEK RETRY COUNT
1124 002310 000000 CKSMRT: .WORD 0      :CHECK SUM RETRY COUNT
1125 002312 000000 CRCBRT: .WORD 0      :CRC BAD RETRY COUNT
1126 002314 000000 CRCERT: .WORD 0      :CRC ERR RETRY COUNT
1127 002316 000000 DATART: .WORD 0      :DATA RETRY COUNT
1128 002320 000000 DARDRT: .WORD 0      :DATA READ RETRY COUNT
1129 002322 000000 DAWTRT: .WORD 0      :DATA WRITE RETRY COUNT
1130 002324 000000 READRT: .WORD 0      :READ RETRY COUNT
1131 002326 000000 WTRT: .WORD 0        :WRITE RETRY COUNT
1132 002330 000000 DDERCT: .WORD 0     :D.D. ERR RETRY COUNT
1133          :-----
```


1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1214
1215
1216
1217
1218
1225
1226

002346

.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
:

1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246 002354 012737 004506 002402
1247 002362 013737 002334 002074
1248 002370
1249 002372 000207
1250
1251 002374
1252 (1) 002374 000000
1253 (1) 002376 000000
1254 (1) 002400 000000
1255 (1) 002402 000000
1256 002404
1257 002444 005737 002272
1258 002450 001452
1259 002452
1260 002522
1261 002572 005037 002272
1262 002576 005037 002604
1263 002602 000207
1264
1265 002604 000000
1266
1267 002606 040445 052440 044516
1268 002663 045 020101 051105
1269 002744 040445 052040 051124
1270 003034
1271
1272
1273

.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..
:--

```

.SBTTL - MOD U.SFT.ERR - ERROR REPORT

```

-----
ERROR: MOV #NONE,ERRBLK ;SETUP ERROR BLOCK CODE
      MOV UNIT,L$LUN ;SETUP LUN FOR PRINT
      ERROR ;
      RETURN ;
-----

```

```

ERRTBL
ERRTYP:: .WORD 0
ERRNBR:: .WORD 0
ERRMSG:: .WORD 0
ERRBLK:: .WORD 0
-----

```

.SBTTL - MOD U.PRT.ERR - PRINT ERRORS

```

-----
PRTErr: PRINTB #IDENT1,UNIT,CSRUT,ESRUT,CMD
IFAUP: TST PRTECD ;IF ERR CODE FLAG
      BEQ ENDUP ;SET, THEN
      PRINTX #XER1,<B,XERUT>,<B,WC>,<B,CTK0>,<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 CTRK0=%D2%A. CTRK1=%D2%A./
XER2: .ASCIZ /%A TTRK=%D2%A. TSEC=%D2%A. SFTSTAT=%03%. BTRK=%D2%A.%N/
      .EVEN
:MOD U.PRT.ERR ----- END MODULE -----

```

1276
 1277
 1278
 1279
 1280 003034 105737 033544
 1281 003040 001424
 1282 003042 013701 033544
 1283 003046 042701 177400
 1284 003052 006201
 1285 003054 006201
 1286 003056 062701 003114
 1287 003062 011137 003114
 1288 003066
 1289 003106 105037 033544
 1290 003112 000207
 1291
 1292
 1293 003114 000000
 1294
 1295
 1296 003116 003170
 1297 003120 003236
 1298 003122 003304
 1299 003124 003332
 1300 003126 003400
 1301 003130 003451
 1302 003132 003477
 1303 003134 003555
 1304 003136 003603
 1305 003140 003660
 1306 003142 003714
 1307 003144 003773
 1308 003146 004021
 1309 003150 004107
 1310 003152 004153
 1311 003154 004207
 1312 003156 004254
 1313 003160 004311
 1314 003162 004360
 1315 003164 004413
 1316 003166 004442
 1317

```

.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
        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
-----

```

```

1320
1321 003170 040445 020040 037040
1322 003236 040445 020040 037040
1323 003304 040445 020040 037040
1324 003332 040445 020040 037040
1325 003400 040445 020040 037040
1326 003451 045 020101 020040
1327 003477 045 020101 020040
1328 003555 045 020101 020040
1329 003603 045 020101 020040
1330 003660 040445 020040 037040
1331 003714 040445 020040 037040
1332 003773 045 020101 020040
1333 004021 045 020101 020040
1334 004107 045 020101 020040
1335 004153 045 020101 020040
1336 004207 045 020101 020040
1337 004254 040445 020040 037040
1338 004311 045 020101 020040
1339 004360 040445 020040 037040
1340 004413 045 020101 020040
1341 004442 040445 020040 037040
1342
1343
1344
1345
1346
1347 004506
1348 004506
1349
1366 004510
1367 004510 004737 004536
1368 004514
1369
1370 004516
1371 004534 000207
1372
1373 004536
1374 004556 000207
1375

```

```

-----
EC1: .ASCIZ /%A >NO HOME ON INITIALIZE-DRV #0.%N/
EC2: .ASCIZ /%A >NO HOME ON INITIALIZE-DRV #1.%N/
EC3: .ASCIZ /%A >ILL ERR CODE.%N/
EC4: .ASCIZ /%A >TRIED TO ACCESS A TRACK > 76.%N/
EC5: .ASCIZ /%A >HOME FOUND BEFORE DESIRED TRACK.%N/
EC6: .ASCIZ /%A >ILL ERR CODE.%N/
EC7: .ASCIZ /%A >52 HEADERS PASSED & SECTOR NOT FOUND.%N/
EC10: .ASCIZ /%A >ILL ERR CODE.%N/
EC11: .ASCIZ /%A >NO SEPCLOCK SEEN IN 40 MICROSECONDS.%N/
EC12: .ASCIZ /%A >PREAMBLE NOT FOUND.%N/
EC13: .ASCIZ /%A >PREAMBLE FOUND BUT NO ID MARK IN TIME.%N/
EC14: .ASCIZ /%A >ILL ERR CODE.%N/
EC15: .ASCIZ /%A >GOOD TRACK ADDRESS HEADER NOT=SELECTED TRACK.%N/
EC16: .ASCIZ /%A >TOO MANY TRIES FOR AN IDAM.%N/
EC17: .ASCIZ /%A >NO DATA AM IN TIME.%N/
EC20: .ASCIZ /%A >CRC ERROR ON READING SECTOR.%N/
EC21: .ASCIZ /%A >UNASSIGNED ERR CODE.%N/
EC22: .ASCIZ /%A >R-W ELECT. FAILED MAINT. TEST.%N/
EC23: .ASCIZ /%A >WORD CNT OVERFLOW.%N/
EC24: .ASCIZ /%A >DENSITY ERROR.%N/
EC25: .ASCIZ /%A >SET DENSITY WRONG KEY WORD.%N/
.EVEN
-----

```

```

-----
.SBTTL - ERROR PRINT CALLS/MSG CALLS
-----

```

```

-----
BGNMSG NONE
ENDMSG
-----
BGNMSG PRTB1
CALL PRTB1S
ENDMSG
-----
PRTB0S: PRINTB R1
RETURN
-----
PRTB1S: PRINTB R1,R2
RETURN
-----

```

1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1394
1400
1407
1413
1420
1429
1437
1443
1444
1451
1457
1458
1459 004560 012700 000001
1460 004564 063700 004646
1461 004570 063700 004650
1462 004574 042700 170000
1463 004600 000241
1464 004602 006100
1465 004604 006100
1466 004606 010037 004646
1467 004612 005000
1468 004614 013700 004650
1469 004620 006000
1470 004622 006000
1471 004624 063700 004646
1472 004630 042700 170000
1473 004634 010037 004650
1474 004640 010037 004652
1475 004644 000207
1476
1477 004646 000000
1478 004650 000000
1479 004652 000000
1480

.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,R0
          ADD   RAN1,R0
          ADD   RAN2,R0
          BIC   #170000,R0
          CLC
          ROL   R0
          ROL   R0
          MOV   R0,RAN1
          CLR   R0
          MOV   RAN2,R0
          ROR   R0
          ROR   R0
          ADD   RAN1,R0
          BIC   #170000,R0
          MOV   R0,RAN2
          MOV   R0,RANUM
          RTS   PC
-----
RAN1:    0
RAN2:    0
RANUM:   0
-----

```


1483
1484
1485
1486 004654 000240
1487 004656 005037 004754
1488 004662 032737 000001 004752
1489 004670 001014
1490 004672 032737 000002 004752
1491 004700 001004
1492 004702 052737 000001 004754
1493 004710 000417
1494 004712 052737 000004 004754
1495 004720 000413
1496 004722 032737 000002 004752
1497 004730 001004
1498 004732 052737 000002 004754
1499 004740 000403
1500 004742 052737 000010 004754
1501 004750 000207
1502
1503 004752 000000
1504 004754 000000
1505
1506
1507
1508
1509
1510 004756 013705 021426
1511 004762 005004
1512 004764 032705 000001
1513 004770 001003
1514 004772 006205
1515 004774 005204
1516 004776 000772
1517 005000 010437 005024
1518 005004 006304
1519 005006 010437 002240
1520 005012 062704 002336
1521 005016 011437 002334
1522 005022 000207
1523
1524 005024 000000
1525

```

.SBTTL - MOD U.A.1 - CONVERSION UUT CODE --> SUTPTR
;-----
CVUTST: NOP ;
CLR SUTCV ;CLEAR SYS UNDER TEST CONVERTED
BIT #1,CVUNIT ;IF DRIVE #0.
BNE 2$ ;SELECTED, THEN
BIT #2,CVUNIT ;IF UNIT #0 OR RXXX SIDE #0,
BNE 1$ ;THEN
BIS #1,SUTCV ;SET FOR UNIT CODE=00 IN SUT WORD
BR ENDCVT ;BR TO END
1$: BIS #4,SUTCV ;ELSE, SET FOR UNIT CODE=10 IN SUT WORD
BR ENDCVT ;BR TO END
2$: BIT #2,CVUNIT ;IF UNIT #0 OR RXXX SIDE #0,
BNE 3$ ;THEN
BIS #2,SUTCV ;SET FOR UNIT CODE=01 IN SUT WORD
BR ENDCVT ;BR TO END
3$: BIS #10,SUTCV ;ELSE, SET FOR UNIT CODE=11 IN SUT WORD
ENDCVT: RTS PC ;RETURN
;-----
CVUNIT: 0 ;UNIT CODE TO BE CONVERTED
SUTCV: 0 ;SYS UNDER TEST AS CONVERTED
;MOD U.A.1 ----- END MODULE -----
.SBTTL - MOD U.A.2 - CONVERSION SUTPTR --> UUT CODE
;-----
CVSTUT: MOV SUTPTR,R5 ;SAVE SUT POINTER IN R5
CLR R4 ;CLEAR R4 (RESET UNIT CODE)
1$: BIT #1,R5 ;IF LSB R5
BNE 2$ ;EQUALS 1 ,THEN BR TO 2$
ASR R5 ;SHIFT RIGHT R5
INC R4 ;INCREMENT R4
BR 1$ ;BR TO 1$
2$: MOV R4,UNITST ;THEN R4 CONTAINS UUT CODE
ASL R4 ;DOUBLE UNIT CODE FOR ADR
MOV R4,UUTOFF ;SET UUT OFFSET
ADD #U00,R4 ;GET UUT UNIT# FOR PRINT
MOV (R4),UNIT ;SET UNIT=PRINT UNIT#
RTS PC ;RETURN
;-----
UNITST: 0 ;
;MOD 2.0A ----- END MODULE -----

```

1528
 1529
 1530
 1531 005026 000240
 1532 005030 013705 002236
 1533 005034 012737 000001 002272
 1534 005042 012737 000017 005136
 1535 005050 053737 002242 005136
 1536 005056 013715 005136
 1537 005062 013701 002236
 1538 005066 062701 000002
 1539 005072 013737 002236 025332
 1540 005100 012737 000200 025330
 1541 005106 004737 025230
 1542 005112 032715 000200
 1543 005116 001004
 1544 005120 052737 040007 002274
 1545 005126 000402
 1546 005130 012711 033544
 1547 005134 000207
 1548
 1549 005136 000000
 1550
 1551
 1552 005140
 1553

```

.SBTTL - MOD U.DEV.REC - DEVICE READ ERROR CODE
-----
RDERCD: NOP
MOV UUTADR,R5 ;SET R5 = UUT ADDRESS
MOV #1,PRTECD ;SET PRINT ERROR CODE FLAG
MOV #17,RECCMD ;SET UUT EXTENDED ERROR CODE
BIS DEN,RECCMD ;SET DEN FOR CMD
MOV RECCMD,(R5) ;SEND CMD TO UUT
MOV UUTADR,R1 ;GET UUT ADDR
ADD #2,R1 ;CAL DATA ADR
MOV UUTADR,CSRADR ;SET CSR ADR
MOV #TRBIT,RDYWD ;SET "TR" BIT TEST
CALL DELAY ;CALL DELAY MODULE-WAIT FOR TR
IAREC: BIT #200,(R5) ;IF TR
BNE LAREC ;NOT SET
BIS #40007,ERRSY ;THEN SET "TR" ERR ON FUNCTION
BR XREC ;BR TO END MOD
LAREC: MOV #XERUUT,(R1) ;SEND BASE ADDR FOR EXTEND ERR CODE
XREC: RETURN ;RETURN
-----
RECCMD: 0 ;COMMAND WORD USED IN THIS MODULE
-----

```

ENDMOD

1566
 1567
 1604
 1605 005140
 1606
 1607
 1608
 1509
 1610
 1611
 1612
 1613 005140
 1614 005140 000240
 1615 005142 012737 006074 005504
 1616 005150 012737 006274 005506
 1617 005156 004737 005414
 1618 005162 004737 005646
 1619 005166 012737 006105 005504
 1620 005174 012737 006253 005506
 1621 005202 004737 005414
 1622 005206 000240
 1623 005210 005037 005636
 1624 005214 005037 005644
 1625 005220 012702 007354
 1626 005224 012701 006360
 1627 005230 012737 000023 005640
 1628 005236 004737 005510
 1629 005242 012737 006200 005504
 1630 005250 012737 006253 005506
 1631 005256 004737 005414
 1632 005262 000240
 1633 005264 012737 000001 005636
 1634 005272 012737 000001 005644
 1635 005300 012702 007604
 1636 005304 012701 006327
 1637 005310 012737 000027 005640
 1638 005316 012737 006327 005642
 1639 005324 004737 005510
 1640 005330 012737 006225 005504
 1641 005336 012737 006253 005506
 1642 005344 004737 005414
 1643 005350 005037 005636
 1644 005354 012737 000001 005644
 1645 005362 012702 010070
 1646 005366 012737 000115 005640
 1647 005374 012737 006343 005642
 1648 005402 004737 005510
 1649 005406
 1650
 1651 005410 000000
 1652 005412 000000
 1653

```

.TITLE MISCELLANEOUS SECTIONS
.SBTTL REPORT CODING SECTION

      BGNMOD

:++
: THE REPORT CODING SECTION CONTAINS THE
: 'PRINTS' CALLS THAT GENERATE STATISTICAL REPORTS.
:--

-----
:
: BGNRPT
REPORT: NOP
:
: MOV #PT20SP,PRT1 :SETUP CTR HDR
: MOV #PTUNT2,PRT2 :
: CALL PRTHDR :PRINT HEADER
: CALL PRTCTR :PRINT SEQ CTR
: 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.,LINEC : SET # OF LINES TO PRINT
: CALL PRTDAT :PRINT DATA
: MOV #PTEC,PRT1 :SETUP HEADER PART 2
: 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 PRTDAT :PRINT DATA
: MOV #PTTK,PRT1 :SETUP HEADER PART 3
: MOV #PTUNT1,PRT2 :
: CALL PRTHDR :PRINT HEADER
: CLR LINECT :
: MOV #1,PRNUM :
: MOV #TKXX,R2 :SETUP DATA PART 3
: MOV #77.,LINES :
: MOV #PTTKN,LINTYP :
: CALL PRTDAT :PRINT DATA PART 3

ENDRPT: ENDRPT

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

```

1656
 1657
 1658 005414 005003
 1659 005416 013705 005504
 1660 005422 004737 006030
 1661 005426 012737 002336 005410
 1662 005434 012737 000004 005412
 1663 005442 005777 177742
 1664 005446 100407
 1665 005450 017737 177734 006026
 1666 005456 013705 005506
 1667 005462 004737 006002
 1668 005466 062737 000002 005410
 1669 005474 005337 005412
 1670 005500 001360
 1671 005502 000207
 1672
 1673 005504 000000
 1674 005506 000000
 1675
 1676
 1677
 1678
 1679 005510 000240
 1680 005512 005737 005644
 1681 005516 001410
 1682 005520 013737 005636 006026
 1683 005526 013705 005642
 1684 005532 004737 006002
 1685 005536 000403
 1686 005540 012105
 1687 005542 004737 006030
 1688 005546 012737 002336 005410
 1689 005554 012737 000004 005412
 1690 005562 012237 006026
 1691 005566 005777 177616
 1692 005572 100404
 1693 005574 012705 006243
 1694 005600 004737 006002
 1695 005604 062737 000002 005410
 1696 005612 005337 005412
 1697 005616 001361
 1698 005620 005237 005636
 1699 005624 023737 005640 005636
 1700 005632 101327
 1701 005634 000207
 1702
 1703 005636 000000
 1704 005640 000000
 1705 005642 000000
 1706 005644 000000
 1707

```

.SBTTL - PRINT REPORT HEADER
-----
PRTHDR: CLR R3
MOV PRT1,R5 ;SETUP 1ST PART OF HEADER PRINT
CALL PREPT2 ;PRINT 1ST PART
MOV #UT00,UTTST ;GET BEGIN ADR OF UNITS-->TESTED FLAGS
MOV #4,UTCNT ;SET UNIT COUNTER
1$: TST @UTTST ;IF UNIT TESTED FLAG
BMI 2$ ;NOT=-1, THEN
MOV @UTTST,PAR ;SET UNIT TESTED # FOR PRINT
MOV PRT2,R5 ;SET UNIT MSG
CALL PREPT1 ;PRINT UNIT #
2$: ADD #2,UTTST ;ADVANCE ADR OF UNIT TESTED FLAG
DEC UTCNT ;DECREMENT UNIT COUNT
BNE 1$ ;IF UNIT COUNT=0, THEN
RTS PC ;RETURN
-----
PRT1: 0
PRT2: 0
-----
.SBTTL - PRINT REPORT DATA
-----
PRDAT: NOP
1$: TST PRNUM ;IF MODE
BEQ 2$
MOV LINECT,PAR ;SETUP LINE # TO PRINT
MOV LINTYP,R5 ;SETUP LINE TYPE TO PRINT
CALL PREPT1 ;PRINT LINE #
BR 3$
2$: MOV (R1)+,R5 ;SETUP LOG TITLE ADR
CALL PREPT2 ;PRINT LOG TITLES
3$: MOV #UT00,UTTST ;GET UNIT # FOR PRINT
MOV #4,UTCNT ;SETUP UNIT COUNT
4$: MOV (R2)+,PAR ;SETUP DATA TO PRINT
TST @UTTST ;IF UNIT # NOT = -1
BMI 5$ ;THEN
MOV #PTDAT1,R5 ;SETUP TO PRINT
CALL PREPT1 ;PRINT DATA
5$: ADD #2,UTTST ;SETUP TO CK NEXT UNIT
DEC UTCNT ;DECREMENT UNIT COUNT
BNE 4$ ;IF DONE ALL UNITS THEN
INC LINECT ;INCREMENT LINE COUNT
CMP LINES,LINECT ;IF DONE ALL
BHI 1$ ;LINES, THEN
RTS PC ;RETURN
-----
LINECT: 0 ;LINE COUNTER
LINES: 0 ;# OF LINES TO PRINT
LINTYP: 0 ;LINE PRINT TYPE.
PRNUM: 0 ;PRINT MODE
-----

```

```

1710
1711
1712
1713 005646 000240
1714 005650 005037 005640
1715 005654 012702 007314
1716 005660 012705 006116
1717 005664 004737 006002
1718 005670 012737 002336 005410
1719 005676 012737 000004 005412
1720 005704 005777 177500
1721 005710 100410
1722 005712 062702 000002
1723 005716 011204
1724 005720 014203
1725 005722 012705 006315
1726 005726 004737 006050
1727 005732 062737 000002 005410
1728 005740 062702 000004
1729 005744 005337 005412
1730 005750 001355
1731 005752 005237 005640
1732 005756 022737 000002 005640
1733 005764 001405
1734 005766 012702 007334
1735 005772 012705 006147
1736 005776 000732
1737 006000 000207
1738
  
```

```

.SBTTL - PRINT READ/WRITE SECTOR COUNTERS
-----
PRTCTR: NOP
CLR LINES ;CLEAR LINE COUNTER
MOV #READSC,R2 ;GET ADDRESS OF READ SECTOR CTR
MOV #PTRDSC,R5 ;SETUP READ SECTORS MSG
1$: CALL PREPT1 ;CALL PRINT REPORT-MSG
MOV #UT00,UTTST ;GET UNIT # FOR PRINT
MOV #4,UTCNT ;SETUP UNIT COUNT
2$: TST @UTTST ;IF UNIT #
BMI 5$ ;NOT=-1, THEN
ADD #2,R2 ;INCREMENT ADR TO UPPER WORD
MOV (R2),R4 ;SETUP DATA UPPER PART FOR PRINT
MOV -(R2),R3 ;SETUP DATA LOWER PART FOR PRINT
MOV #PTFMN1,R5 ;SETUP TO PRINT DATA
CALL PREPT3 ;PRINT DATA
5$: ADD #2,UTTST ;SETUP TO CK NEXT UNIT
ADD #4,R2 ;SET ADR TO NEXT CTR
DEC UTCNT ;DECREMENT UNIT COUNT
BNE 2$ ;IF DONE THIS LINE, THEN
INC LINES ;INCREMENT LINE CTR
CMP #2,LINES ;DO WHILE LINE CTR
BEQ 6$ ;EQUALS <2
MOV #WRITSC,R2 ;GET ADDRESS OF WRITE SECTOR CTR
MOV #PTWTSC,R5 ;SETUP WRITE SECTORS MSG
BR 1$ ;BR TO WRITE SECTORS SECTION
6$: RETURN ;RETURN
-----
  
```

1741
 1742
 1743 006002
 1744 006024 000207
 1745
 1746 006026 000000
 1747
 1748
 1749
 1750
 1751
 1752
 1753
 1754
 1755 006030
 1756 006046 000207
 1757
 1758
 1759
 1760
 1761
 1762
 1763
 1764
 1765 006050
 1766 006072 000207
 1767
 1768 006074 047045 047045 051445
 1769 006105 045 022516 022516
 1770 006116 047045 040445 020043
 1771 006147 045 022516 021501
 1772 006200 047045 047045 040445
 1773 006225 045 022516 022516
 1774 006243 045 020101 022440
 1775 006253 045 030523 040445
 1776 006274 051445 022462 052501
 1777 006315 045 031123 047445
 1778 006327 045 022516 031117
 1779 006343 045 022516 030523
 1780 006360
 1781

```

.SBTTL - PRINT REPORT TYPE 1
-----
PREPT1: PRINTS R5,PAR
        RTS    PC
-----
PAR:    0
-----

```

```

.SBTTL - PRINT REPORT TYPE 2
-----
PREPT2: PRINTS R5
        RTS    PC
-----

```

```

.SBTTL - PRINT REPORT TYPE 3
-----
PREPT3: PRINTS R5,R4,R3
        RETURN
-----
PT20SP: .ASCIZ  /%N%N%S20/
PT19SP: .ASCIZ  /%N%N%S19/
PTRDSC: .ASCIZ  /%N%A# SECTOR READS (8)=/
PTWTSC: .ASCIZ  /%N%A# SECTOR WRITES (8)=/
PTEC:   .ASCIZ  /%N%N%AERR%N%ACODE# /
PTTK:   .ASCIZ  /%N%N%ATRACK# /
PTDAT1: .ASCIZ  /%A %D6/
PTUNT1: .ASCIZ  /%S1%AUNIT#%D1%S1/
PTUNT2: .ASCIZ  /%S2%AUNIT#%D1%S5/
PTFMN1: .ASCIZ  /%S2%06%05/
PTECN:  .ASCIZ  /%N%02%A0%S3/
PTTKN:  .ASCIZ  /%N%S1%D2%S3/
        .EVEN
-----

```

1784
 1785 006360 006426
 1786 006362 006455
 1787 006364 006504
 1788 006366 006533
 1789 006370 006562
 1790 006372 006611
 1791 006374 006640
 1792 006376 006667
 1793 006400 006716
 1794 006402 006745
 1795 006404 006774
 1796 006406 007023
 1797 006410 007052
 1798 006412 007101
 1799 006414 007130
 1800 006416 007157
 1801 006420 007206
 1802 006422 007235
 1803 006424 007264
 1804
 1805
 1806
 1807 006426 047045 040445 044103
 1808 006455 045 022516 043101
 1809 006504 047045 040445 047516
 1810 006533 045 022516 044501
 1811 006562 047045 040445 047111
 1812 006611 045 022516 051501
 1813 006640 047045 040445 051103
 1814 006667 045 022516 041501
 1815 006716 047045 040445 042522
 1816 006745 045 022516 053501
 1817 006774 047045 040445 040504
 1818 007023 045 022516 042101
 1819 007052 047045 040445 051110
 1820 007101 045 022516 044101
 1821 007130 047045 040445 051110
 1822 007157 045 022516 044101
 1823 007206 047045 040445 051110
 1824 007235 045 022516 044101
 1825 007264 047045 040445 051110
 1826 007314
 1827

```

-----
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
-----

```

```

-----
PRID01: .ASCIZ /%N%ACHECK SUM: /
PRID02: .ASCIZ /%N%AFILL-EMP BUFF LOG:/
PRID03: .ASCIZ /%N%AND ERR BIT: /
PRID04: .ASCIZ /%N%AINTER-NO DONE ERR:/
PRID05: .ASCIZ /%N%AINTERRUPT ERR: /
PRID06: .ASCIZ /%N%ASEEK: /
PRID07: .ASCIZ /%N%ACRC ERR: /
PRID08: .ASCIZ /%N%ACRC BAD: /
PRID09: .ASCIZ /%N%AREAD ERR: /
PRID10: .ASCIZ /%N%AWRITE ERR: /
PRID11: .ASCIZ /%N%ADATA ERR: /
PRID12: .ASCIZ /%N%ADEL. DATA ERR: /
PRID13: .ASCIZ /%N%AHRD SEEK: /
PRID14: .ASCIZ /%N%AHRD CRC ERR: /
PRID15: .ASCIZ /%N%AHRD CRC BAD: /
PRID16: .ASCIZ /%N%AHRD READ: /
PRID17: .ASCIZ /%N%AHRD WRITE: /
PRID18: .ASCIZ /%N%AHRD DATA: /
PRID19: .ASCIZ /%N%AHRD DEL. DATA ERR:/
        .EVEN
-----

```

1830			.SBTTL - STATISTICAL TABLES		
1831			;		
1832			-----		
1833	007314	000010	READSC: .BLKW 8.		;READ SECTOR COUNTER
1834	007334	000010	WRITSC: .BLKW 8.		;WRITE SECTOR COUNTER
1835	007354	000004	CKSML: .BLKW 4		;CKSUM LOG
1836	007364	000004	BUFERL: .BLKW 4		;FILL/EMPTY BUFFER ERROR LOG
1837	007374	000004	NOERL: .BLKW 4		;NO ERR BIT LOG
1838	007404	000004	UKINT: .BLKW 4		;INTERRUPT - NO DONE LOG
1839	007414	000004	INTER: .BLKW 4		;INTERRUPT ERR
1840	007424	000004	SEK: .BLKW 4		;SEEK ERR
1841	007434	000004	CRC: .BLKW 4		;CRC ERR
1842	007444	000004	CRCBAD: .BLKW 4		;CRC BAD ERR
1843	007454	000004	RD: .BLKW 4		;READ ERR
1844	007464	000004	WRT: .BLKW 4		;WRITE ERR
1845	007474	000004	DATA: .BLKW 4		;DATA ERR
1846	007504	000004	DLDTERR: .BLKW 4		;DEL DATA ERR
1847	007514	000004	HSEK: .BLKW 4		;HARD SEEK ERR
1848	007524	000004	HCRC: .BLKW 4		;HARD CRC ERR
1849	007534	000004	HCRCBD: .BLKW 4		;HARD CRC BAD ERR
1850	007544	000004	HRD: .BLKW 4		;HARD READ ERR
1851	007554	000004	HWRT: .BLKW 4		;HARD WRITE ERR
1852	007564	000004	HDATA: .BLKW 4		;HARD DATA ERR
1853	007574	000004	HDD: .BLKW 4		;HARD DEL DATA ERR
1854	007604	000132	ECLOG: .BLKW 90.		;ERROR CODE LOG
1855	010070	000464	TKXX: .BLKW 308.		;TRACK ERR LOG
1856			;		
1857	011240	000000	ENDST: .WORD 0		;END TABLE
1858					
1870					
1871			.EVEN		
1872					
1873			.SBTTL LOAD DEVICE PROTECTION		
1874			;		
1875	011242		BGNPROT		
1876	011242	000000	.WORD 0		;RX CSR - HARDWARE P-TABLE OFFSET
1877	011244	177777	.WORD -1		;DON'T CARE
1878	011246	000004	.WORD 4		;RX DRIVER-HARDWARE P-TABLE OFFSET
1879	011250		ENDPROT		
1880			;		

1883
 1884
 1885
 1886
 1887
 1888
 1889 011250
 1890 011250 005037 002266
 1895 011254
 1896 011262
 1897 011270
 1898 011272 052737 000001 002266
 1899 011300 000507
 1900 011302
 1901 011310
 1902 011312 005037 002220
 1903 011316 005037 002222
 1904 011322 005037 002224
 1905 011326 005037 002226
 1906 011332 005037 002232
 1907 011336 023727 002012 000004
 1908 011344 003051
 1909 011346
 1910 011354
 1911 011356 052737 000002 002266
 1912 011364 005037 002270
 1913 011370 012737 177777 002334
 1914 011376 012737 177777 002336
 1915 011404 012737 177777 002340
 1916 011412 012737 177777 002342
 1917 011420 012737 177777 002344
 1918 011426 062737 000001 002334
 1919 011434 023737 002012 002334
 1920 011442 001426
 1921 011444
 1922 011456
 1923 011460 000240
 1924 011462 004737 011656
 1925 011466 000757
 1926 011470
 1927 011510 012737 000001 002270
 1928 011516
 1929 011520
 1930 011546 005737 002226
 1931 011552 001413
 1932 011554
 1933 011602
 1953
 1954 011604 000000
 1955
 1956 011606 047045 040445 047117
 1957
 1958

```

.SBTTL INITIALIZE SECTION
:*****
: THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
: AT THE BEGINNING OF EACH PASS.
:-----
INIT:  BGNINIT
      CLR  FLAGS           ;CLEAR ALL FLAGS
      RFLAGS FLGDRS       ;GET 'DRS' FLAGS
      REDEF #EF.PWR       ;IF POWER FAIL FLAG IS
      BNCOMPLETE 1$       ;SET, THEN
      BIS  #POWERF,FLAGS  ;SET POWER FAIL FLAG
      BR   FIN            ;BR TO 'FIN'
1$:   REDEF #EF.START     ;IF START FLAG
      BNCOMPLETE 2$       ;SET, THEN
      CLR  UOADR          ;CLEAR SYS UO ADDRESS
      CLR  U1ADR          ;CLEAR SYS U1 ADDRESS
      CLR  UOVECTOR       ;CLEAR SYS UO VECTOR
      CLR  U1VECTOR       ;CLEAR SYS U1 VECTOR
2$:   CLR  SUT            ;CLEAR SYS UNDER TST WORD
      CMP  LSUNIT,#4      ;IF 4 UNITS OR LESS SELECTED
      BGT  INITER         ;THEN
      REDEF #EF.RESTART   ;IF RESTART FLAG
      BNCOMPLETE SETUP    ;SET, THEN
      BIS  #RESTAR,FLAGS  ;SET RESTART FLAG
      CLR  ABORT          ;CLEAR ABORT FLAG
      MOV  #-1,UNIT       ;RESTORE UNIT # CTR
      MOV  #-1,UT00       ;RESET UNIT#1
      MOV  #-1,UT01       ;RESET UNIT#2
      MOV  #-1,UT10       ;RESET UNIT#3
      MOV  #-1,UT11       ;RESET UNIT#4
1$:   ADD  #1,UNIT        ;INCREMENT TO NEXT UNIT
      CMP  LSUNIT,UNIT    ;IF LOGICAL UNIT & UNIT
      BEQ  FIN            ;NOT YET EQUAL, THEN
      GPHARD UNIT,PLOC    ;GET HARDWARE P-TABLE
      BNCOMPLETE 1$      ;IF P-TABLE AVAILABLE, THEN
      NOP
      JSR  PC,UNPKHP      ;CALL UNPACK HARDWARE P-TABLE
      BR   1$            ;BR TO BEGIN DO
INITER: PRINTF #INTER1    ;PRINT "TOO MANY UNITS"
      MOV  #1,ABORT       ;SET ABORT
      DOCLN              ;DO CLEAN UP
FIN:  SETVEC UOVECTOR,#INTH0,#PRI07 ;SET SYS UO VECTOR
      TST  U1VECTOR       ;IF SYS U1 VECTOR
      BEQ  2$            ;NOT=0, THEN
1$:   SETVEC U1VECTOR,#INTH1,#PRI07 ;SET SYS U1 VECTOR
2$:   ENDINIT
:-----
PLOC: .WORD 0
:-----
INTER1: .ASCIZ /%N%ONLY FOUR UNITS ALLOWED, START OVER/
      .EVEN
:-----

```

```

1961 .SBTTL - MOD I.1 - UNPACK HARDWARE P-TABLES
1962 ;-----
1963
1964 011656 000240 UNPKHP: NOP ;
1965 011660 005037 012330 CLR UNT ;CLEAR UNT
1966 011664 013701 011604 MOV PLOC,R1 ;SAVE P-TABLE LOCATION
1967 011670 005737 002334 IFAI1: TST UNIT ;IF UNIT
1968 011674 001005 BNE IFBI1 ;IS ZERO
1969 011676 012137 002220 MOV (R1)+,UOADR ;LOAD UNIT #0 ADR
1970 011702 012137 002224 MOV (R1)+,UOVECTOR ;LOAD UNIT #0 VECTOR
1971 011706 000426 BR IFBI1 ;BR TO END IF 'A'
1972 011710 021137 002220 IFBI1: CMP (R1),UOADR ;IF THIS ADR
1973 011714 001003 BNE IFCI1 ;EQUALS UNIT #0 ADR
1974 011716 062701 000004 ADD #4,R1 ;INCREMENT TEMP #1 BY 4
1975 011722 000420 BR IFBI1 ;BR TO END IF 'A'
1976 011724 005737 002222 IFCI1: TST U1ADR ;IF U1 ADDRESS
1977 011730 001005 BNE IFDI1 ;NOT LOADED PREVIOUSLY
1978 011732 012137 002222 MOV (R1)+,U1ADR ;LOAD UNIT#1 ADR
1979 011736 012137 002226 MOV (R1)+,U1VECTOR ;LOAD UNIT #1 VECTOR
1980 011742 000405 BR EICI1 ;BR TO END IF 'C'
1981 011744 021137 002222 IFDI1: CMP (R1),U1ADR ;IF UNIT ADR
1982 011750 001153 BNE ELDI1 ;EQUALS UNIT #1 ADR
1983 011752 062701 000004 ADD #4,R1 ;THEN ADD 4 TO TEMP #1
1984 011756 012737 000001 012330 EICI1: MOV #1,UNIT ;SET UNT=1
1985 011764 005737 002172 IFEI1: TST RXXX ;IF RXXX
1986 011770 001445 BEQ IFII1 ;THEN
1987 011772 005711 IFFI1: TST (R1) ;IF DRIVE #0
1988 011774 001021 BNE IFHI1 ;THEN
1989 011776 062701 000002 IFGI1: ADD #2,R1 ;ADD 2 TO TEMP #1
1990 012002 005711 TST (R1) ;IF SIDE #0 SELECTED
1991 012004 001006 BNE ELGI1 ;THEN
1992 012006 052737 000001 002232 BIS #BIT0,SUT ;SET SIDE #0, DRIVE #0
1993 012014 005037 012326 CLR UNTCOD ;CLEAR UNIT CODE
1994 012020 000501 BR EIFI1 ;BR TO END IF 'F'
1995 012022 052737 000004 002232 ELGI1: BIS #BIT2,SUT ;SET SIDE #1, DRIVE #0
1996 012030 012737 000002 012326 MOV #2,UNTCOD ;SET UNIT CODE = 10
1997 012036 000472 BR EIFI1 ;BR TO END IF 'F'
1998 012040 062701 000002 IFHI1: ADD #2,R1 ;ADD 2 TO TEMP #1
1999 012044 005711 TST (R1) ;IF SIDE #0 SELECTED
2000 012046 001007 BNE ELHI1 ;THEN
2001 012050 052737 000002 002232 BIS #BIT1,SUT ;SET SIDE #0, DRIVE #1
2002 012056 012737 000001 012326 MOV #1,UNTCOD ;SET UNIT CODE = 01
2003 012064 000457 BR EIFI1 ;BR TO END IF 'F'
2004 012066 052737 000010 002232 ELHI1: BIS #BIT3,SUT ;SET SIDE #1, DRIVE #1
2005 012074 012737 000003 012326 MOV #3,UNTCOD ;SET UNIT CODE = 11
2006 012102 000450 BR EIFI1 ;BR TO END IF 'F'
2007 012104 062701 000002 IFII1: ADD #2,R1 ;ADD 2 TO R1
2008 012110 005711 TST (R1) ;IF SIDE
2009 012112 001056 BNE ELI1 ;EQUALS 0, THEN
2010 012114 162701 000002 IFJI1: SUB #2,R1 ;SUBTRACT 2 FROM TEMP #1
2011 012120 005711 TST (R1) ;IF DRIVE
2012 012122 001020 BNE IFLI1 ;EQUALS ZERO, THEN
2013 012124 005737 012330 IFKI1: TST UNT ;IF UNIT
2014 012130 001006 BNE ELKI1 ;EQUALS ZERO
2015 012132 052737 000001 002232 BIS #BIT0,SUT ;SET UNIT #0, DRIVE #0
2016 012140 005037 012326 CLR UNTCOD ;CLEAR UNIT CODE

```

```
2017 012144 000427  
2018 012146 052737 000004 002232  
2019 012154 012737 000002 012326  
2020 012162 000420  
2021 012164 005737 012330  
2022 012170 001007  
2023 012172 052737 000002 002232  
2024 012200 012737 000001 012326  
2025 012206 000406  
2026 012210 052737 000010 002232  
2027 012216 012737 000003 012326  
2028 012224 012701 002336  
2029 012230 013702 012326  
2030 012234 006302  
2031 012236 060201  
2032 012240 013703 002334  
2033 012244 010311  
2034 012246 000426  
2035 012250  
2036 012274  
2037 012276 000412  
2038 012300  
2039 012324 000207  
2040  
2041 012326 000000  
2042 012330 000000  
2043  
2044 012332 047045 040445 047125  
2045 012432 047045 040445 047125  
2046 012522  
2047
```

```
ELK11: BR EIF11 ;BR TO END IF 'F'  
BIS #BIT2,SUT ;SET UNIT #1, DRIVE #0  
MOV #2,UNTCOD ;SET UNIT CODE = 10  
BR EIF11 ;BR TO END IF 'F'  
IFL11: TST UNT ;IF UNIT  
BNE ELL11 ;EQUALS 0  
BIS #BIT1,SUT ;SET UNIT #0, DRIVE #1  
MOV #1,UNTCOD ;SET UNIT CODE = 01  
BR EIF11 ;BR TO END IF 'F'  
ELL11: BIS #BIT3,SUT ;SET UNIT #1, DRIVE #1  
MOV #3,UNTCOD ;SET UNIT CODE = 11  
EIF11: MOV #UT00,R1 ;GET BEGINING OF UNIT CODE TABLE  
MOV UNTCOD,R2 ;GET UNIT CODE  
ASL R2 ;DOUBLE R2 FOR ADDRESSING  
ADD R2,R1 ;FIND ADDRESS FOR THIS UNIT CODE  
MOV UNIT,R3 ;GET LOGICAL UNIT#  
MOV R3,(R1) ;SET USER UNIT# FOR PRINT OUT  
BR ENDI1 ;BR TO END MOD  
ELI11: PRINTF #INMSG2,UNIT ;PRINT 'MUST SELECT RXXX TO SEL SIDE'  
DOCLN  
BR ENDI1 ;BR TO END MOD  
ELDI1: PRINTF #INMSG3,UNIT ;PRINT 'NOT SCHEDULED-TWO BUS ADR ONLY'  
ENDI1: RTS PC ;RETURN  
-----  
UNTCOD: 0 ;UNIT CODE  
UNT: 0 ;UNIT FLAG  
-----  
INMSG2: .ASCIZ /%N%AUNIT#%D1% ANS RXXX EXPANSION TO SELECT SIDE #1->START OVER  
INMSG3: .ASCIZ /%N%AUNIT#%D1% NOT SCHEDULED-TWO BUS ADDRESSSES ONLY%N/  
.EVEN  
:MOD I.1 ----- END MODULE -----
```

```
2050 .SBTTL CLEANUP CODING SECTION
2051
2052 :++
2053 : THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
2054 : AT THE END OF EACH PASS.
2055 :--
2056
2057 012522 BGNCLN
2058 012522 000240 NOP
2065 012524 CLRVEC UOJECT
2066 012532 005737 002226 TST U1VECT
2067 012536 001403 BEQ 2$
2068 012540 CLRVEC U1VECT
2069 012546 2$: BRESET
2070 012550 ENDCLN
2071
2083
2084 .EVEN
```

```

2087
2088
2089 012552
2090 012552 005737 002220
2091 012556 001447
2092 012560 012703 002336
2093 012564 013702 002220
2094 012570 004737 012774
2095 012574 005737 002270
2096 012600 001436
2097 012602 005737 002336
2098 012606 100403
2099 012610
2100 012616 005737 002340
2101 012622 100403
2102 012624
2103 012632 005737 002172
2104 012636 001417
2105 012640 012703 002342
2106 012644 005737 002342
2107 012650 100403
2108 012652
2109 012660 005737 002344
2110 012664 100440
2111 012666
2112 012674 000434
2113 012676 005737 002222
2114 012702 001425
2115 012704 012703 002342
2116 012710 013702 002222
2117 012714 004737 012774
2118 012720 005737 002270
2119 012724 001420
2120 012726 005737 002342
2121 012732 100403
2122 012734
2123 012742 005737 002344
2124 012746 100403
2125 012750
2126 012756 005737 002220
2127 012762 001001
2128 012764
2129 012766 005037 002270
2130 012772
2131

```

.SBITL AUTO DROP SECTION

```

-----
:
:BGNAUTO
IAATDP: TST UOADR ;IF SYS UNIT 0 ADDRESS
:BEQ IDATDP ;NOT=0, THEN
:MOV #UT00,R3 ;SETUP R3 = ADR OF SELECTED UNIT
:MOV UOADR,R2 ;GET SYS UNIT 0 ADDRESS
:CALL ADRTST ;CALL ADDRESSING TEST
IBATDP: TST ABORT ;IF ABORT FLAG
:BEQ IDATDP ;SET, THEN
IGATDP: TST UT00 ;IF UT00 SELECTED
:BMI IHATDP ;THEN
:DODU UT00 ;DROP UNIT 00
IHATDP: TST UT01 ;IF UT01 SELECTED
:BMI ICATDP ;THEN
:DODU UT01 ;DROP UNIT 01
ICATDP: TST RXXX ;IF RXXX DEVICE
:BEQ IDATDP ;THEN
:MOV #UT10,R3 ;SETUP R3 = ADR OF SELECTED UNIT
IIATDP: TST UT10 ;IF UT10 SELECTED
:BMI IJATDP ;THEN
:DODU UT10 ;DROP UNIT 10
IJATDP: TST UT11 ;IF UT11 SELECTED
:BMI XATDP ;THEN
:DODU UT11 ;DROP UNIT 11
:BR XATDP ;BR TO EXIT
IDATDP: TST U1ADR ;IF SYS UNIT 1 ADDRESS
:BEQ IFATDP ;NOT=0, THEN
:MOV #UT10,R3 ;SETUP R3 = ADR OF SELECTED UNIT
:MOV U1ADR,R2 ;GET SYS UNIT 1 ADDRESS
:CALL ADRTST ;CALL ADDRESSING TEST
IEATDP: TST ABORT ;IF ABORT FLAG
:BEQ XATDP ;SET, THEN
IKATDP: TST UT10 ;IF UT10 SELECTED
:BMI ILATDP ;THEN
:DODU UT10 ;DROP UNIT 10
ILATDP: TST UT11 ;IF UT11 SELECTED
:BMI IFATDP ;THEN
:DODU UT11 ;DROP UNIT 11
IFATDP: TST UOADR ;IF SYS UNIT 0 ADDRESS
:BNE XATDP ;EQUALS 0, THEN
XATDP: DOCLN ;DO CLEAN
:CLR ABORT ;CLEAR ABORT FLAG
:ENDAUTO
-----

```

2134
 2135
 2136
 2137
 2138
 2139
 2140
 2141
 2142
 2143
 2144
 2145
 2146
 2147
 2148
 2149
 2150
 2151 012774 000240
 2152 012776 005037 002270
 2153 013002
 2154 013030 011201
 2155 013032
 2156 013040 005737 002270
 2157 013044 001426
 2158 013046 012701 013144
 2159 013052 012337 002074
 2160 013056 100005
 2161 013060 011337 002074
 2162 013064 100002
 2163 013066 005037 002074
 2164 013072 012737 000620 002376
 2165 013100 012737 013124 002400
 2166 013106 012737 004510 002402
 2167 013114 005037 002374
 2168 013120
 2169 013122 000207
 2170
 2171 013124 042101 051104 051505
 2172 013144 040445 041040 051525
 2173 013200 040445 044440 052116
 2174 013260

```

.SBTTL - TEST 0: ADDRESSING TEST
-----
BGNSUB
: IF FUNCTION TEST
: THEN-SETUP TEST
:   SETUP BUS TRAPS
:   READ RXCSR
:   RESET BUS TRAPS
:   IF TRAP
:     THEN-SET SYSTEM FATAL FLAG
:     CALL FUNCTION TEST ERROR
:     REPORT BUS TRAP ON RXCSR
:   ENDIF
: ENDIF
ENDSUB
-----

ADRTST: NOP
:
CLR ABORT :CLEAR ABORT FLAG
SETVEC #BTRP4,#TRAP,#PRI07
MOV (R2),R1 :READ RXCSR
CLRVEC #BTRP4
TST ABORT :IF ABORT FLAG
BEQ 2$ :SET, THEN
MOV #TRPMS1,R1 :SET TRAP MESSAGE
MOV (R3)+,L$LUN :IF UNIT
BPL 1$ :NOT SELECTED, THEN
MOV (R3),L$LUN :IF NEXT UNIT
BPL 1$ :NOT SELECTED, THEN
CLR L$LUN :CLEAR UNIT
1$: MOV #400.,ERRNBR :SETUP ERR NBR = ADR ERR
MOV #TOMSG,ERRMSG :SETUP ERROR MSG
MOV #PRTB1,ERRBLK :SETUP ERROR BLK
CLR ERRTYP :SETUP ERR TYP = SYS FTL
ERROR :CALL ERROR
2$: RETURN :RETURN
-----

TOMSG: .ASCIZ /ADDRESSING TEST/
TRPMS1: .ASCII /%A BUS TRAP AT ADDRESS:%06%N/
: .ASCIZ /%A INTERFACE BAD OR NOT SET TO ABOVE ADDRESS%N/
: .EVEN
  
```



```

2197          .SBTTL  DROP UNIT SECTION
2198
2199          :++
2200          : THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
2201          : TO NO LONGER BE TESTED.
2202          :--
2203
2209
2210 013274          BGNDU
2211
2212 013274 010037 013436      MOV     R0,UNITDP      ;GET LOGICAL UNIT #
2213 013300 005002          CLR     R2              ;LET R2=UNIT CODE# & UNIT COUNT /CLEAR IT!
2214 013302 012701 002336      MOV     #UT00,R1      ;GET BEGIN UNIT CODE ADDRESS
2215 013306 023721 013436      1$:    CMP     UNITDP,(R1)+ ;IF USER UNIT#
2216 013312 001417          BEQ     2$              ;IS = UNIT CODE - UNIT#
2217 013314 005202          INC     R2              ;INCREMENT UNIT CODE# & UNIT COUNT
2218 013316 022702 000005      CMP     #5,R2         ;IF MAX # OF UNITS
2219 013322 101371          BHI     1$              ;EXCEEDED, THEN
2220 013324          PRINTF #DUMSG2,UNITDP ;PRINT UNIT# NOT FOUND
2221 013350 000431          BR     3$              ;BR TO EXIT
2222 013352 012741 177777      2$:    MOV     #-1,-(R1) ;DESELECT UNIT
2223 013356 010237 004752      MOV     R2,CVUNIT     ;SET UNIT CODE FOR CONVERSION
2224 013362 004737 004654      CALL   CVUTST         ;CALL MOD U.A.1 CONVERT UNIT# TO SUT CODE
2225 013366 013737 004754 013440  MOV     SUTCV,SUTDRP  ;SET SUT DROP CODE = SUT CONVERTED CODE
2226 013374 043737 013440 002232  BIC     SUTDRP,SU;    ;DROP UNIT SPEC IN SUTDRP
2227 013402 043737 013440 002230  BIC     SUTDRP,SDD    ;CLEAR UNIT SPEC IN SUT DROP
2228 013410          PRINTF #DUMSG1,UNITDP
2229
2230 013434      3$:    ENDDU
2231          :-----
2232 013436 000000      UNITDP: 0              ;UNIT TO BE DROPPED
2233 013440 000000      SUTDRP: 0             ;SYS UNDER TST, DROP BIT
2234          :-----
2235 013442 047045 040445 042040  DUMSG1: .ASCIZ  /%N% DROP UNIT#%D1% FROM TEST%N/
2236 013503 045 022516 020101  DUMSG2: .ASCIZ  /%N% COULD NOT DROP UNIT#%D1% -NOT SELECTED%N/
2237          :-----
2238
2250
2251          .EVEN
  
```


2254
2255
2256
2257
2258
2259
2260
2261
2262 013562
2263
2269
2270 013562
2271
2283
2284

.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

ENDAU

.EVEN

2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331

```
.TITLE HARDWARE TESTS  
.SBTTL TEST 1: RX02 SS PERF EXERCISER  
:++  
: TEST TO EXERCISE RX02/XX SYSTEM  
:--  
.SBTTL MOD 0.0 - EXERCISE A SYSTEM  
-----  
: BGNTST  
: BGND0  
: : BGNSUB  
: : : INITIALIZE (LOCATIONS, ETC.)  
: : : CALL MOD 1.0  
: : : ENDSUB  
: : IF ERR SYS=1  
: : : THEN-  
: : : : CALL MOD 4.0  
: : : ENDF  
: : IF ABORT=0  
: : : THEN-  
: : : : BGND0  
: : : : : BGNSUB  
: : : : : : CALL MOD 2.0  
: : : : : IF ERR SYS NOT=0  
: : : : : : THEN-  
: : : : : : : CALL MOD 4.0  
: : : : : : IF ABORT=0  
: : : : : : : THEN-  
: : : : : : : : CALL MOD 3.0  
: : : : : : : ENDF  
: : : : : : : ELSE-  
: : : : : : : : CALL MOD 3.0  
: : : : : : : ENDF  
: : : : : : : CK LOOP  
: : : : : : : ENDSUB  
: : : : : DO UNTIL ABORT=1 OR EXCMP=1  
: : : ENDF  
: : DO UNTIL SWREG BIT#15 NOT SET  
: : IF ABORT=1  
: : : THEN-  
: : : : DO CLEAN UP  
: : : ELSE-  
: : : : DO REPORT  
: : ENDF  
: : ENDTST  
-----
```

```
2333 013564          BGNTST
2334 013564 000240   CONTRL: NOP
2335 013566          BG00:  BGNSUB
2336 013570 005037 014020   CLR      EXCMP
2337 013574 005037 002270   CLR      ABORT
2338 013600 012737 000001 014016   MOV      #1,INITL
2339 013606 005037 002304   CLR      RETRY
2340 013612 005037 002230   CLR      SDD
2341 013616 005037 002274   CLR      ERRSY
2342 013622 005037 002276   CLR      ERRTY
2343 013626 005037 002246   CLR      CSRUUT
2344 013632 005037 002250   CLR      ESRUUT
2345 013636 005037 033544   CLR      XERUUT
2346 013642 005037 002332   CLR      CMD
2347 013646 005037 023330   CLR      WDOT
2348 013652 012737 000001 021426   MOV      #1,SUTPTR
2349 013660 004737 014022   CALL     G1SYEX
2350 013664          ENDSUB
2351 013666 005737 022274   IA00:   TST      ERRSY
2352 013672 001402          BEQ      IB00
2353 013674 004737 032466   CALL     OTSYER
2354 013700 005737 002270   IB00:   TST      ABORT
2355 013704 001030          BNE      UG00
2356 013706          BC00:   BGNSUB
2357 013710 004737 020676   CALL     SCSYEX
2358 013714 005737 002274   ID00:   TST      ERRSY
2359 013720 001410          BEQ      LD00
2360 013722 004737 032466   CALL     OTSYER
2361 013726 005737 002270   IE00:   TST      ABORT
2362 013732 001005          BNE      ED00
2363 013734 004737 032444   CALL     OTEXCM
2364 013740 000402          BR       ED00
2365 013742 004737 032444   LD00:   CALL     OTEXCM
2366 013746          ED00:   CKLOOP
2367 013750          ENDSUB
2368 013752 005737 002270   UC00:   TST      ABORT
2369 013756 001007          BNE      IF00
2370 013760 005737 014020          TST      EXCMP
2371 013764 001750          BEQ      BC00
2372 013766 032737 100000 002204   UG00:   BIT      #100000,SWREG
2373 013774 001274          BNE      BG00
2374 013776 005737 002270   IF00:   TST      ABORT
2375 014002 001402          BEQ      LF00
2376 014004          DOCLN
2377 014006 000401          BR       END00
2378 014010          LF00:   DORPT
2379 014012          END00: EXIT    TST
2380
-----
2381 014016 000000   INITL: 0
2382 014020 000000   EXCMP: 0
2383
:MOD 0.0 ----- END MODULE -----
:BEGIN SUB TEST
:CLEAR EXERCISE COMPLETE
:CLEAR ABORT FLAG
:SET INITIALIZE FLAG
:CLEAR RETRY FLAGS
:CLEAR SYS DRIVES DONE
:CLEAR SYSTEM ERROR FLAGS
:CLEAR DEVICE ERROR FLAGS
:CLEAR UUT CSR
:CLEAR UUT ESR
:CLEAR UUT TEST ERROR REG
:CLEAR COMMAND PRINT WORD
:CLEAR COMMAND WORD
:PRESET SYS UNDER TST PTR
:CALL MOD 1.0 GET SYS EXER.
:END SUB TEST
:IF ERR SYS
:NOT=0, THEN
:CALL MOD 4.0 - O/P SYSTEM ERPOR
:IF ABORT
:NOT SET, THEN
:BEGIN SUB TEST
:CALL MOD 2.0 - SCHEDULE SYSTEM EXERCISE
:IF ERR SYSTEM
:NOT=0, THEN
:CALL MOD 4.0 - O/P SYSTEM ERROR
:IF ABORT
:NOT SET, THEN
:CALL MOD 3.0 - O/P SYSTEM EXERCISE COMPLETE
:BR TO END 'D'
:CALL MOD 3.0 - O/P SYSTEM EXERCISE COMPLETE
:CHECK LOOP ON ERROR
:END SUB TEST
:DOUNTIL ABORT
:OR
:EXERCISE COMPLETE
:SET
:DOUNTIL SWREG BIT#15
:NOT SET
:IF ABORT
:SET, THEN
:DO CLEAN UP
:BR TO END
:DO REPORT
:EXIT TEST
:INITIALIZE POINTERS FLAG
:EXERCISE COMPLETE FLAG
```

```

2386
2387
2388
2389
2390 014022 000240
2391 014024 032737 000001 002266
2392 014032 001002
2393 014034 004737 014076
2394 014040 032737 040000 002204
2395 014046 001002
2396 014050 004737 014216
2397 014054 004737 017302
2398 014060 042737 040000 002274
2399 014066 005037 014074
2400 014072 000207
2401
2402 014074 000001
2403
2404
2405
2406
2407
2408
2409
2410
2411 014076 000240
2412 014100 032737 000001 002204
2413 014106 001404
2414 014110 012737 000200 002252
2415 014116 000403
2416 014120 012737 000100 002252
2417 014126 013737 002206 020650
2418 014134 013737 002210 020652
2419 014142 032737 000002 002204
2420 014150 001404
2421 014152 012737 000010 002244
2422 014160 000402
2423 014162 005037 002244
2424 014166 032737 000001 002204
2425 014174 001404
2426 014176 012737 000400 002242
2427 014204 000402
2428 014206 005037 002242
2429 014212 000240
2430 014214 000207
2431

.SBTTL MOD 1.0 - GET SYSTEM EXERCISE
-----
GTSYEX: NOP
IFB10: BIT #POWERF,FLAGS ;IF POWER FLAG
      BNE IFA10 ;NOT SET, THEN
      JSR PC,GTEXCD ;CALL GET EXERCISE CONDITION
IFA10: BIT #40000,SWREG ;IF NO INITIALIZE
      BNE ELA10 ;NOT SET, THEN
      JSR PC,GTSYS ;CALL GET SYSTEM TO EXERCISE
ELA10: JSR PC,GTEX ;CALL GET EXERCISE
      BIC #BIT14,ERRSY ;CLEAR ANY TIME OUT ERRORS ALREADY REPORTED
      CLR FIRST ;CLEAR FIRST PASS FLAG
      RTS PC ;RETURN
-----
FIRST: 1 ;FIRST PASS FLAG
;MOD 1.0 ----- END MODULE -----

.SBTTL MOD 1.1 - GET EXERCISE CONDITIONS
-----
GTEXCD: NOP
IFA11: BIT #1,SWREG ;IF SET FOR DOUBLE DENSITY
      BEQ ELA11 ;THEN
      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)
      BIT #2,SWREG ;IF DEL DATA SET
      BEQ ELB11 ;THEN
      MOV #10,DEL DAT ;SET DEL DATA MODE
      BR IFC11 ;BR TO END IF 'B'
ELB11: CLR DEL DAT ;CLEAR DEL DATA MODE
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 -----

```

2434
2435
2436
2437
2438
2439 014216
2440 014220 004737 014626
2441 014224 012737 000040 025330
2442 014232 013737 002220 025332
2443 014240 004737 025230
2444 014244 032777 000040 165746
2445 014252 001006
2446 014254 012737 016167 016130
2447 014262 004737 016016
2448 014266 000442
2449 014270 012777 040000 165722
2450 014276 012737 000040 025330
2451 014304 013737 002220 025332
2452 014312 004737 025230
2453 014316 032777 000040 165674
2454 014324 001006
2455 014326 012737 016235 016130
2456 014334 004737 016016
2457 014340 000415
2458 014342 012737 000002 015276
2459 014350 012737 000001 015302
2460 014356 005037 015300
2461 014362 013704 002220
2462 014366 004737 014742
2463 014372 000412
2464 014374 005737 002172
2465 014400 001404
2466 014402 042737 000017 002232
2467 014410 000403
2468 014412 042737 000003 002232
2469 014420 005737 002172
2470 014424 001401
2471 014426 000463
2472 014430 032737 000014 002232
2473 014436 001457
2474 014440 004737 014702
2475 014444 032777 000040 165550
2476 014452 001441
2477 014454 012777 040000 165540
2478 014462 012737 000040 025330
2479 014470 013737 002222 025332
2480 014476 004737 025230
2481 014502 032777 000040 165512
2482 014510 001416
2483 014512 012737 000004 015302
2484 014520 012737 000002 015276
2485 014526 012737 000002 015300
2486 014534 013704 002222
2487 014540 004737 014742
2488 014544 000414
2489 014546 012737 016235 016130

.SBTTL MOD 1.2 - GET SYSTEM TO EXERCISE

GTSYS: BRESET ;ISSUE BUS RESET
CALL GPSUN0 ;CALL GET PRINTABLE SYSTEM 0 UNIT #
MOV #DNBIT,RDYWD ;SET READY WORD = DONE
MOV UOADR,CSRADR ;SET ADDRESS
CALL DELAY ;CALL MOD - DELAY FOR DONE
IFA12: BIT #DNBIT,@UOADR ;IF UNIT #0 DONE BIT
BNE ELA12 ;NOT SET THEN
MOV #INTER2,ITMSG ;SET PRINT MSG#
CALL ITERR ;INITIALIZE ERR-U0-NO DONE BIT
BR EIA12 ;BR TO END IF 'A'
ELA12: MOV #40000,@UOADR ;ELSE-ISSUE PROG INIT TO UO
MOV #DNBIT,RDYWD ;SET READY WORD = DONE
MOV UOADR,CSRADR ;SET TEST ADDRESS
CALL DELAY ;CALL MOD - DELAY FOR DONE
IFB12: BIT #DNBIT,@UOADR ;IF UNIT #0 DONE BIT
BNE ELB12 ;NOT SET THEN
MOV #INTER3,ITMSG ;SET PRINT MSG#
CALL ITERR ;INITIALIZE ERR-U0, NO DONE BIT
BR EIA12 ;BR TO END IF 'A'
ELB12: MOV #2,UNTCNT ;SET # DRVS TO CK
MOV #1,SUTPOS ;SET POSITION IN SUT TO TEST = 1
CLR UNTCO ;SET UUT CODE = 0
MOV UOADR,R4 ;SET TEMP #4 = UO ADDRESS
CALL CKDVAV ;CALL MOD 1.2.1 - CK DRIVE STATUS
BR IFC12 ;BR TO IF 'C'
EIA12: TST RXXX ;IF RXXX
IFH12: BEQ ELH12 ;THEN
BIC #17,SUT ;CLEAR RXXX UO SELECTED DRIVES
BR IFC12 ;BR TO IF 'C'
ELH12: BIC #3,SUT ;CLEAR RX02 UO SELECTED DRIVES
IFC12: TST RXXX ;IF RXXX
BEQ IFD12 ;THEN
BR IFG12 ;BR TO IF 'G'
IFD12: BIT #14,SUT ;IF U1
BEQ IFG12 ;SELECTED THEN
CALL GPSUN1 ;CALL GET PRINTABLE SYSTEM 1 UNIT #
IFE12: BIT #DNBIT,@U1ADR ;IF U1 DONE BIT
BEQ ELE12 ;SET THEN
MOV #40000,@U1ADR ;INITIALIZE DEVICE U1
MOV #DNBIT,RDYWD ;SET READY WORD = DONE BIT
MOV U1ADR,CSRADR ;SET TEST ADR
CALL DELAY ;CALL MOD - WAIT FOR DONE
IFF12: BIT #DNBIT,@U1ADR ;IF U1 DONE BIT
BEQ ELF12 ;SET THEN
MOV #4,SUTPOS ;SET POSITION IN SUT = 4
MOV #2,UNTCNT ;SET # DRVS TO CK = 2
MOV #2,UNTCO ;SET UUT CODE = 2
MOV U1ADR,R4 ;SET TEMP #4 = U1 ADR
CALL CKDVAV ;CALL MOD 1.2.1 - CK DRIVE STATUS
BR IFG12 ;BR TO IF 'G'
ELF12: MOV #INTER3,ITMSG ;SET MSG#-U1-'NO DONE BIT-PROG INT''

2490 014554 000403
 2491 014556 012737 016167 016130
 2492 014564 004737 016016
 2493 014570 042737 000014 002232
 2494 014576 005737 002232
 2495 014602 001007
 2496 014604 012701 016305
 2497 014610 004737 004516
 2498 014614 012737 000001 002270
 2499 014622 000240
 2500 014624 000207
 2501
 2502
 2503
 2504
 2505 014626 005037 002334
 2506 014632 005737 002336
 2507 014636 100404
 2508 014640 013737 002336 002334
 2509 014646 000414
 2510 014650 005737 002340
 2511 014654 100404
 2512 014656 013737 002340 002334
 2513 014664 000405
 2514 014666 005737 002172
 2515 014672 001402
 2516 014674 004737 014702
 2517 014700 000207
 2518
 2519
 2520
 2521
 2522 014702 005037 002334
 2523 014706 005737 002342
 2524 014712 100404
 2525 014714 013737 002342 002334
 2526 014722 000406
 2527 014724 005737 002344
 2528 014730 100403
 2529 014732 013737 002344 002334
 2530 014740 000207
 2531

```

BR      EIE12      ;BR TO END IF 'E'
EIE12:  MOV      #INTER2,ITMSG ;SET MSG#-U1-'NO DONE BIT-BUS INIT'
EIE12:  CALL     ITERR      ;INIT ERR
        BIC      #14,SUT    ;CLEAR SYS 1 FROM TEST
IFG12:  TST      SUT        ;IF SYSTEM UNDER TEST
        BNE     ELG12      ;EQUALS 0, THEN
        MOV     #INTER4,R1  ;SETUP PRINT - 'NO SYS TO TEST'
        CALL   PRTBOS      ;CALL PRINT BASIC-0 ARG
        MOV     #1,ABORT    ;SET ABORT FLAG
ELG12:  NOP
        RTS      PC        ;RETURN
;MOD 1.2 ----- END MODULE -----

```

.SBTTL - MOD 1.2.U.1 - GET PRINTABLE SYSTEM 0 UNIT #

```

GPSUN0: CLR      UNIT      ;SET UNIT=0
        TST     UT00      ;IF UT00
        BMI     2$        ;VALID, THEN
        MOV     UT00,UNIT  ;SETUP UNIT FOR PRINT
        BR     XPSUN0     ;BR TO EXIT
2$:     TST     UT01      ;IF UT01
        BMI     3$        ;VALID, THEN
        MOV     UT01,UNIT  ;SETUP UNIT FOR PRINT
        BR     XPSUN0     ;BR TO EXIT
3$:     TST     RXXX      ;IF RXXX
        BEQ     XPSUNC    ;THEN
        CALL   GPSUN1    ;CALL GET PRINTABLE SYSTEM 1 UNIT #
XPSUN0: RETURN          ;RETURN

```

.SBTTL - MOD 1.2.U.2 - GET PRINTABLE SYSTEM 1 UNIT #

```

GPSUN1: CLR      UNIT      ;SET UNIT=0
1$:     TST     UT10      ;IF UT10
        BMI     2$        ;VALID, THEN
        MOV     UT10,UNIT  ;SETUP UNIT FOR PRINT
        BR     XPSUN1     ;BR TO EXIT
2$:     TST     UT11      ;IF UT11
        BMI     XPSUN1    ;VALID, THEN
        MOV     UT11,UNIT  ;SETUP UNIT FOR PRINT
XPSUN1: RETURN          ;RETURN

```

```

2534      .SBTTL MOD 1.2.1 - CK DRIVE AVAILABLE
2535      -----
2536 014742 010437 015272 CKDVAV: MOV R4,ITCSAD ;SAVE C & S ADR
2537 014746 062704 000002 ADD #2,R4 ;SET DATA BUFFER ADR
2538 014752 010437 015274 MOV R4,ITDBAD ;SAVE DB ADR
2539 014756 000240 BDA121: NOP ;
2540 014760 033737 015302 002232 IFA121: BIT SUTPOS,SUT ;IF THIS UNIT SUT & SUT
2541 014766 001521 BEQ EIA121 ;EQUAL, THEN
2542 014770 BGNSEG ;BEGIN SEGMENT-TO LOOP ON ERROR
2543 014772 013701 015300 MOV UNTC, R1 ;SAVE UNIT CODE #
2544 014776 006301 ASL R1 ;DOUBLE UNIT CD FOR ADR
2545 015000 062701 002336 ADD #UT00,R1 ;FIND ADR UNIT#
2546 015004 011137 002334 MOV (R1),UNIT ;SET UNIT# FOR PRINT
2547 015010 032737 000001 015300 IFB121: BIT #1,UNTC ;IF DRIVE #1 SET IN UNIT CODE
2548 015016 001407 BEQ ELB121 ;THEN
2549 015020 012737 000033 015266 MOV #33,INTCMD ;SET READ STATUS DRV #1
2550 015026 012737 000001 015270 MOV #1,DRIVEN ;SET PRINT FOR DRV #1
2551 015034 000405 BR EIB121 ;BR TO END IF 'B'
2552 015036 012737 000013 015266 ELB121: MOV #13,INTCMD ;SET READ STATUS DRV #0
2553 015044 005037 015270 CLR DRIVEN ;SET PRINT FOR DRIVE #0
2554 015050 013777 015266 000214 EIB121: MOV INTCMD,@ITCSAD ;EXECUTE READ STATUS ON DRIVE AT TEMP #4
2555 015056 013737 015272 025332 MOV ITCSAD,CSRADR ;PASS DOWN ADRS
2556 015064 012737 000040 025330 MOV #DNBIT,RDYWD ;PASS DOWN 'DONE' BIT TO TEST
2557 015072 004737 025230 CALL DELAY ;CALL MOD - DELAY FOR DONE BIT
2558 015076 032777 000010 000170 IFH121: BIT #10,@ITDBAD ;IF AC LOW BIT
2559 015104 001404 BEQ IFC121 ;SET, THEN
2560 015106 012737 017200 016130 MOV #ITER3,ITMSG ;SET MSG# - 'AC LOW'
2561 015114 000436 BR EIC121 ;BR TO END IF 'C'
2562 015116 032777 000200 000150 IFC121: BIT #200,@ITDBAD ;IF DRV RDY BIT
2563 015124 001004 BNE IFI121 ;NOT SET, THEN
2564 015126 012737 016334 016130 MOV #ITMSG1,ITMSG ;SET MSG# - 'NO DRIVE READY'
2565 015134 000426 BR EIC121 ;BR TO END IF 'C'
2566 015136 032777 004000 000126 IFI121: BIT #RX2BIT,@ITCSAD ;IF CSR RX02 BIT
2567 015144 001004 BNE IFD121 ;NOT SET, THEN
2568 015146 012737 016521 016130 MOV #ITMSG5,ITMSG ;SET MSG # 'NOT CAP. OF DOUBLE DENS. OPS.'
2569 015154 000416 BR EIC121 ;BR TO END IF 'C'
2570 015156 005737 002172 IFD121: TST RXXX ;IF UNIT IS TO BE TESTED AS RXXX
2571 015162 001421 BEQ EID121 ;THEN
2572 015164 032737 000002 015300 IFE121: BIT #2,UNTC ;IF SIDE #1
2573 015172 001415 BEQ EID121 ;SELECTED
2574 015174 032777 000002 000072 IFF121: BIT #2,@ITDBAD ;IF SIDE #1
2575 015202 001011 BNE EID121 ;NOT READY, THEN
2576 015204 012737 016357 016130 MOV #ITMSG2,ITMSG ;SET MSG# - 'NO SIDE RDY'
2577 015212 004737 016016 EIC121: CALL ITERR ;CALL INITIALIZE ERROR
2578 015216 BGNSEG ;END SEGMENT-TO LOOP ON ERROR
2579 015220 004737 016064 CALL ITDROP ;CALL DROP UNIT
2580 015224 000402 BR EIA121 ;BR TO ENDIF 'A'
2581 015226 004737 015306 EID121: CALL REFORV ;CALL REFORMAT DRIVE DENSITY
2582 015232 006137 015302 EIA121: ROL SUTPOS ;MOVE SELECT BIT TO TEST SYS UNDER TEST
2583 015236 005337 015276 DEC UNTCNT ;DECREMENT UNIT COUNT
2584 015242 005237 015300 INC UNTC ;INCREMENT UNIT UNDER TEST CODE
2585 015246 005737 015276 DUA121: TST UNTCNT ;DO
2586 015252 001402 BEQ END121 ;UNTIL
2587 015254 000137 014756 JMP BDA121 ;ALL UNITS DONE
2588 015260 000240 END121: NOP ;
2589 015262 000207 RTS PC ;RETURN

```

HARDWARE TESTS MACY11 30(1046) 29-MAR-82 15:57 PAGE 42-1
CZRXC.P11 29-MAR-82 14:53 MOD 1.2.1 - CK DRIVE AVAILABLE

SEQ 0051

2590

;------

2593
 2594 015264 000000
 2595 015266 000000
 2596 015270 000000
 2597 015272 000000
 2598 015274 000000
 2599 015276 000000
 2600 015300 000000
 2601 015302 000000
 2602 015304 000000
 2603

```

:-----:
REFCMD: 0      ;REFORMAT COMMAND
INTCMD: 0      ;INITIAL COMMAND WORD
DRIVEN: 0      ;DRIVE NUMBER
ITCSAD: 0      ;INITIAL C & S ADR
ITDBAD: 0      ;INITIAL DATA BUFFER ADR
UNTCNT: 0      ;UNIT COUNT
UNTCO: 0      ;UNIT CODE
SUTPOS: 0      ;SYS UNDER TST POSITION
FORMCK: 0      ;FORMAT CK FLAG
:-----:

```

```

2606 .SBTTL MOD 1.2.1.1 - REFORMAT DRIVE DENSITY
2607 :-----
2608
2609 015306 033737 015302 002232 REFDRV: BIT SUTPOS,SUT ;IF UNIT SELECTED IN
2610 015314 001003 BNE IA1211 ;SYS UNDER TEST
2611 015316 000137 016012 JMP X1211 ;THEN
2612 015322 BGNSEG ;BEGIN SEGMENT-FOR LOOP ON ERROR
2613 015324 032737 000001 002204 IA1211: BIT #1,SWREG ;IF DOUBLF DENSITY
2614 015332 001417 BEQ IC1211 ;SET, THEN
2615 015334 032777 000040 177732 IB1211: BIT #40,@ITDBAD ;IF DISKETTE IS NOT DOUBLE DENSITY
2616 015342 001011 BNE LB1211 ;THEN
2617 015344 012737 016401 016130 MOV #ITMSG3,ITMSG ;SET MSG# DSK SGL DEN
2618 015352 004737 016102 CALL ITPRNT ;CALL PRINT -
2619 015356 012737 000400 015264 MOV #BIT8,REFCMD ;SET REFORMAT CMD TO DOUBLE DENSITY
2620 015364 000417 BR ID1211 ;BR TO IF 'D'
2621 015366 000137 016012 LB1211: JMP X1211 ;ELSE BR TO END
2622 015772 032777 000040 177674 IC1211: BIT #40,@ITDBAD ;IF DISKETTE
2623 015400 001002 BNE 1$ ;IS NOT SINGLE DENSITY, THEN
2624 015402 000137 016012 JMP X1211 ;-
2625 015406 012737 016576 016130 1$: MOV #ITMSG6,ITMSG ;SET MSG# DSK DBL DEN
2626 015414 004737 016102 CALL ITPRNT ;CALL PRINT -
2627 015420 005037 015264 CLR REFCMD ;SET REFORMAT CMD TO SINGLE DENSITY
2628 015424 ID1211: MANUAL ;IF MANUAL INTERVENTION
2629 015426 BNCOMPLETE LD1211 ;IS ALLOWED,THEN
2630 015430 GMANIL FCKMSG,FORMCK,1,YES
2631 015444 005737 015304 IE1211: TST FORMCK ;IF REFORMAT
2632 015450 001544 BEQ LE1211 ;OK, THEN
2633 015452 005037 015304 CLR FORMCK ;CLEAR REFORMAT CK
2634 015456 052737 000011 015264 BIS #11,REFCMD ;SET REFORMAT CMD
2635 015464 032737 000001 015300 IF1211: BIT #1,UNTCO ;IF DRIVE #1
2636 015472 001403 BEQ IG1211 ;SELECTED
2637 015474 052737 000020 015264 BIS #BIT4,REFCMD ;SET DRIVE #1 ON REFORMAT CMD
2638 015502 005737 002172 IG1211: TST RXXX ;IF RXXX
2639 015506 001407 BEQ EG1211 ;DEVICE AND
2640 015510 032737 000002 015300 BIT #2,UNTCO ;SIDE #1
2641 015516 001403 BEQ EG1211 ;SELECTED, THEN
2642 015520 052737 001000 015264 BIS #BIT9,REFCMD ;SET SIDE #1 ON REFORMAT CMD
2643 015526 013777 015264 177536 EG1211: MOV REFCMD,@ITCSAD ;SEND REFORMAT CMD
2644 015534 013737 015272 025332 MOV ITCSAD,CSRADR ;PASS UNIT ADRS
2645 015542 012737 000200 025330 MOV #TRBIT,RDYWD ;PASS 'TR' BIT TO TEST
2646 015550 004737 025230 CALL DELAY ;CALL DELAY
2647 015554 005737 002274 IH1211: TST ERRSY ;IF
2648 015560 001070 BNE LH1211 ;T.O. ERR
2649 015562 012777 000111 177504 MOV #111,@ITDBAD ;SEND VARIFY WORD (ASCII 'I')
2650 015570 013702 002334 MOV UNIT,R2 ;SETUP UNIT # PRT
2651 015574 012701 016753 MOV #ITMSG9,R1 ;SET MSG# WRG DEN REFORMAT
2652 015600 004737 004536 CALL PRTBIS ;CALL PRINT BASIC-1 ARG
2653 015604 013737 015272 025332 MOV ITCSAD,CSRADR ;SET UNIT BUS ADR
2654 015612 012737 000040 025330 MOV #DNBIT,RDYWD ;SET DONE BIT TST
2655 015620 013737 025324 016014 MOV RYDX,SAVDLY ;SAVE NORMAL DELAY MULTIPLIER
2656 015626 012737 001000 025324 MOV #1000,RYDX ;SET DELAY MULT HIGH
2657 015634 004737 025230 CALL DELAY ;DELAY UNTIL DONE OR T. O.
2658 015640 013737 016014 025324 MOV SAVDLY,RYDX ;RESET DELAY MULT
2659 015646 017737 177420 002246 MOV @ITCSAD,CSRUT ;GET UUT CSR
2660 015654 017737 177414 002250 MOV @ITDBAD,ESRUUT ;GET UUT ESR
2661 015662 032777 000040 177402 II1211: BIT #40,@ITCSAD ;IF DONE BIT

```

```
2662 015670 001420  
2663 015672 032777 100000 177372  
2664 015700 001444  
2665 015702 013737 015264 002332  
2666 015710 013737 015272 002236  
2667 015716 004737 005026  
2668 015722 012737 017032 016130  
2669 015730 000407  
2670 015732 012737 017116 016130  
2671 015740 000403  
2672 015742 012737 016454 016130  
2673 015750 004737 016016  
2674 015754 004737 002404  
2675 015760 000411  
2676 015762 012737 016652 016130  
2677 015770 000403  
2678 015772 012737 016704 016130  
2679 016000 004737 016016  
2680 016004  
2681 016006 004737 016064  
2682 016012 000207  
2683  
2684 016014 000000  
2685
```

```
                BEQ      LI1211      ;SET ,THEN  
IJ1211: BIT      #100000,@ITCSAD ;IF ERR BIT NOT SET  
                BEQ      X1211      ;THEN BR TO EXIT  
                MOV      REFCMD,CMD  ;SET COMMAND FOR PRINT  
                MOV      ITCSAD,UUTADR ;SET UUT ADR  
                CALL     RDERCD      ;CALL DEVICE READ ERROR CODE  
                MOV      #ITER1,ITMSG ;ELSE, SET 'ERROR ON REFORMAT' MSG  
                BR       EH1211      ;BR TO END IF 'H'  
LI1211: MOV      #ITER2,ITMSG      ;SET 'NO DONE BIT AFTER REFORMAT' MSG  
                BR       EH1211      ;BR TO END IF 'H'  
LH1211: MOV      #ITMSG4,ITMSG      ;SET MSG# NO 'TR' BIT TIME OUT ERR  
EH1211: CALL     ITERR              ;CALL INITIALIZE ERROR  
                CALL     PRERR       ;CALL PRINT ERR  
                BR       EA1211      ;BR TO END IF 'A'  
LE1211: MOV      #ITMSG7,ITMSG      ;SET MSG# DISK WRG DEN  
                BR       ED1211      ;BR TO END IF 'D'  
LD1211: MOV      #ITMSG8,ITMSG      ;SET MSG# MAN INTERVENTION NOT ALL  
ED1211: CALL     ITERR              ;CALL INITIALIZE ERROR  
EA1211: ENDSEG   ;END SEGMENT-TO LOOP ON ERROR  
                CALL     ITDROP      ;CALL DROP UNIT  
X1211: RTS      PC                  ;RETURN  
-----  
SAVDLY: 0      ;SAVE NORMAL DELAY MULTIPLIER  
-----
```

2688
2689
2690
2691 016016 012737 000310 002376
2692 016024 012737 016132 002400
2693 016032 012737 004506 002402
2694 016040 012737 000001 002374
2695 016046 013737 002334 002074
2696 016054
2697 016056 004737 016102
2698 016062 000207
2699
2700
2701
2702
2703 016064 013737 015302 013440
2704 016072
2705 016100 000207
2706
2707
2708
2709
2710
2711 016102 013702 002334
2712 016106 012701 016153
2713 016112 004737 004536
2714 016116 013701 016130
2715 016122 004737 004516
2716 016126 000207
2717
2718 016130 000000
2719
2720 016132 047111 052111 040511
2721 016153 045 020101 047125
2722 016167 045 026501 026455
2723 016235 045 026501 026455
2724 016305 045 022516 020101
2725 016334 040445 020055 047516
2726 016357 045 026501 047040
2727 016401 045 026501 053440
2728 016454 040445 020055 052042
2729 016521 045 026501 047040
2730 016576 040445 020055 051127
2731 016652 040445 042040 051511
2732 016704 040445 046440 047101
2733 016753 045 020101 047125
2734 017032 040445 020055 051105
2735 017116 040445 020055 047516
2736 017200 040445 020055 041501
2737 017223 040 020040 020040
2738 017302
2739

```
.SBTTL - MOD 1.2.U.3 - INITIALIZE ERROR
-----
ITERR: MOV #200,ERRNBR ;SET ERR NBR = INIT ERR
      MOV #ITERMG,ERRMSG ;
      MOV #NONE,ERRBLK ;
      MOV #1,ERRTYP ;SET ERR TYP = DEV FTL
      MOV UNIT,L$LUN ;SETUP LUN FOR PRINT
      ERROR ;CALL ERROR
      CALL ITPRNT ;CALL INITIALIZE PRINT
      RETURN ;RETURN
-----

.SBTTL - MOD 1.2.U.4 - INITIALIZE DROP
-----
ITDROP: MOV SUTPOS,SUTDRP ;SETUP SYS. UNDER TEST DROP BIT
        DODU UNIT ;DROP THIS UNIT FROM TEST
        RTS PC ;RETURN
-----

.SBTTL - MOD 1.2.U.5 - INITIALIZE PRINT
-----
ITPRNT: MOV UNIT,R2 ;SETUP TO PRINT UNIT #
        MOV #ITERUT,R1 ;SETUP MSG
        CALL PRTB1S ;PRINT BASIC-1 ARG
        MOV ITMSG,R1 ;SETUP TO PRINT MSG
        CALL PRTB0S ;PRINT BASIC-0 ARG
        RTS PC ;RETURN
-----

ITMSG: 0 ;INITIALIZE MSG#
-----

ITERMG: .ASCIZ /INITIALIZE ERROR/
ITERUT: .ASCIZ /%A UNIT#%D1/
INTER2: .ASCIZ /%A---NO DONE BIT AFTER BUS INITIALIZE/
INTER3: .ASCIZ /%A---NO DONE BIT AFTER PROG. INITIALIZE/
INTER4: .ASCIZ /%N% NO SYSTEM TO TEST/
ITMSG1: .ASCIZ /%A- NO DRIVE READY/
ITMSG2: .ASCIZ /%A- NO SIDE READY/
ITMSG3: .ASCIZ /%A- WRONG DENSITY -SINGLE DENSITY DISKETTE/
ITMSG4: .ASCIZ /%A- "TR" BIT AFTER SET DENSITY CMD%N/
ITMSG5: .ASCIZ /%A- NOT CAPABLE OF DOUBLE DENSITY OPERATIONS/
ITMSG6: .ASCIZ /%A- WRONG DENSITY - DOUBLE DENSITY DISKETTE/
ITMSG7: .ASCIZ /%A DISKETTE WRONG DENSITY/
ITMSG8: .ASCIZ /%A MAN. INTERVENTION REQ'D - REFORMAT/
ITMSG9: .ASCIZ /%A UNIT#%D1%-REFORMATTING, DO NOT INTERRUPT%N/
ITER1: .ASCIZ /%A- ERROR BIT SET AFTER REFORMAT COMMAND SEQUENCE%N/
ITER2: .ASCIZ /%A- NO DONE BIT AFTER REFORMAT COMMAND SEQUENCE%N/
ITER3: .ASCIZ /%A- AC LOW BIT SET/
FCKMSG: .ASCIZ / ->REFORMAT DISKETTE - ARE YOU SURE?/
        .EVEN
-----
```

2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760

017302 013737 002200 017750
017310 004737 017376
017314 013737 002202 020654
017322 013737 002206 020650
017330 013737 002210 020652
017336 004737 017752
017342 005737 014074
017346 001007
017350 032737 000040 002204
017356 001406
017360 042737 000040 002204
017366 004737 020656
017372 000240
017374 000207

```
.SBTTL MOD 1.3 - GET EXERCISE  
:-----  
GTEX:  MOV    TSTPAT,PAT      ;GET TEST PATTERN #  
       CALL   STSTPA          ;CALL MOD 1.3.1 SET TEST PATTERN  
       MOV    TRKSEQ,SEQUEN   ;GET TRACK SEQ #  
       MOV    OTDITK,OD       ;GET OUTSIDE DIA. TRK  
       MOV    INDITK,ID       ;GET INSIDE DIA. TRK  
       CALL   STKSEQ          ;CALL MOD 1.3.2 SET TRACK SEQUENCE  
IFB13:  TST    FIRST          ;IF A FIRST PASS  
       BNE    THC13          ;THEN  
IFC13:  BIT    #40,SWREG      ;IF CLEAR STATISTICAL TABLES  
       BEQ    END13          ;IS SELECTED THEN  
       BIC    #40,SWREG      ;CLEAR SELECTED - CLR STAT TABLE  
THC13:  CALL   CLRSTA         ;CALL MOD 1.3.3 - CLEAR STATISTICAL TABLES  
       NOP  
END13:  RTS    PC            ;RETURN  
:MOD 1.3 ----- END MODULE -----
```

```

2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780 017376 042737 000377 017462
2781 017404 005037 017744
2782 017410 005737 017750
2783 017414 001003
2784 017416 012737 000007 017750
2785 017424 013704 017750
2786 017430 005304
2787 017432 006304
2788 017434 150437 017462
2789 017440 012704 034010
2790 017444 013705 002252
2791 017450 006305
2792 017452 062705 034006
2793 017456 162705 000004
2794 017462 000777
2795 017464 000137 017520
2796 017470 000137 017536
2797 017474 000137 017546
2798 017500 000137 017614
2799 017504 000137 017622
2800 017510 000137 017646
2801 017514 000137 017656
2802
2803 017520 005037 017746
2804 017524 004737 017704
2805 017530 005705
2806 017532 001463
2807 017534 000773
2808
2809 017536 112737 000377 017746
2810 017544 000767
2811
2812 017546 112737 000376 017746
2813 017554 000261
2814 017556 012702 000000
2815 017562 103001
2816 017564 005202
2817 017566 004737 017704
2818 017572 005705

```

```

.SBTTL MOD 1.3.1 - SET DATA PATTERN
-----
PAT #      ROUTINE      DATA PATTERN
-----
0          RANDAT      NO PATTERN SPECIFIED (FORCE RANDOM DATA)
1          DATA0      ALL ZEROS
2          DATA1      ALL ONES
3          FLOAT0      FLOAT ZERO THRU ONE'S
4          FLOAT1      FLOAT ONE THRU ZERO'S
5          PAT125      ALTERNATING BITS IN ONE BYTE COMP IN NEXT
6          PAT333      ALTERNATING 1'S PAIR & 0 IN ONE BYTE COMP IN NEX
7          RANDAT      RANDOM

:NOTE. DATA PATTERNS WILL BE MODIFIED SO BYTE #0 WILL CONTAIN TRACK ADDRESS
AND BYTE #1 THE SECTOR ADDRESS IN WHICH THE DATA IS WRITTEN.
THE LAST TWO BYTES CONTAIN THE CHECK SUM NUMBERS.
-----
STSTPA: BIC      #377,@#BRONPT :CLEAR BRANCH OFFSET
        CLR      SUM          :SET UP FOR ACCUMULATION OF CHECK SUM
        TST      PAT          :IF NO PATTERN SPECIFIED FORCE PATTERN 7
        BNE      1$
        MOV      #7,PAT
1$:     MOV      PAT,R4        :GET PATTERN BITS
        DEC      R4           :ADJUST FOR CORRECT OFFSET
        ASL      R4
        BISB     R4,@#BRONPT  :INSERT OFFSET
        MOV      #DATPAT+2,R4 :SET UP ADDRESS OF FIRST BYTE
        MOV      WDCNT,R5     :SETUP WORD COUNT
        ASL      R5           :DOUBLE WORD COUNT FOR ADR
        ADD      #DATPAT,R5   :ADD DATA PATTERN ADR
        SUB      #4,R5        :ADJ. FOR CHECKSUM
BRONPT: BR      .            :BRANCH BY OFFSET SELECTED
        JMP      DATA0      :000 DATA BYTE
        JMP      DATA1      :377 DATA BYTE
        JMP      FLOAT0      :FLOAT A 0 THROUGH ALL 1'S
        JMP      FLOAT1      :FLOAT A 1 THROUGH ALL 0'S
        JMP      PAT125      :125/052 DATA WORD
        JMP      PAT333      :314/063 DATA WORD
        JMP      RANDAT      :RANDOM DATA BYTE
-----
DATA0:  CLR      DATBYT
PG:     JSR      PC,LOAD      :GO LOAD THE DATA BUFFER
        TST      R5          :IF R5
        BEQ     END131       :NOT =0 ,THEN
        BR      PG
-----
DATA1:  MOVB     #377,DATBYT
        BR      PG
-----
FLOAT0: MOVB     #376,DATBYT  :SET UP A ONES FIELD
XPG:   SEC          :SET THE C BIT TO ROTATE THROUGH THE DATA
1$:    MOV      #0,R2        :CLR R2 (CAN'T USE "CLR" AS IT CLEARS "C" BIT)
        BCC     2$           :BR IF THE "C" BIT IS CLEARED
        INC     R2           :SET R2 IF NOT
2$:    JSR      PC,LOAD      :GO LOAD THE DATA BUFFER
        TST      R5          :IF R5

```

```

2819 017574 0C1442          BEQ     END131          ;NOT ZERO THEN
2820 017576 000241          CLC
2821 017600 005702          TST     R2              ;IS R2 NONZERO
2822 017602 001401          BEQ     3$              ;YES, SET THE 'C' BIT
2823 017604 000261          SEC
2824 017606 106137 017746    3$:    ROLB    DATBYT
2825 017612 000761          BR     1$
-----
2826
2827 017614 005037 017746    FLOAT1: CLR    DATBYT
2828 017620 000755          BR     XPG
-----
2829
2830 017622 112737 000125 017746    PAT125: MOVB   #125,DATBYT
2831 017630 004737 017704    XXPG:  JSR    PC,LOAD
2832 017634 005705          TST     R5              ;IF R5
2833 017636 001421          BEQ     END131          ;NOT ZERO THEN
2834 017640 105137 017746    COMB   DATBYT
2835 017644 000771          BR     XXPG
-----
2836
2837 017646 112737 000333 017746    PAT333: MOVB   #333,DATBYT
2838 017654 000765          BR     XPG
-----
2839
2840 017656 004737 004560          RANDAT: JSR    PC,RANGEN ;GET RANDOM NUMBER
2841 017662 113737 004652 0177'6    MOVB   RANUM,DATBYT
2842 017670 004737 017704    JSR    PC,LOAD
2843 017674 005705          TST     R5              ;IF R5
2844 017676 001401          BEQ     END131          ;NOT ZERO THEN
2845 017700 000766          BR     RANDAT
-----
2846
2847 017702 000207          END131: RTS    PC        ;RETURN.
-----
2848
2849
2850
-----
2851 017704 063737 017746 017744    LOAD:  ADD    DATBYT,SUM ;ACCUMULATE THE PATTERN CHECK SUM
2852 017712 113724 017746          MOVB   DATBYT,(R4)+    ;LOAD THE DATA BUFFER
2853 017716 020504          CMP    R5,R4           ;HAVE 124 BYTES BEEN GENERATED
2854 017720 001401          BEQ     1$             ;IF YES, RETURN
2855 017722 000407          BR     ENLDL           ;IF NO, RETURN TO PATTERN GENERATOR
2856 017724 113724 017744    1$:    MOVB   SUM,(R4)+  ;PUT CHECKSUM INTO TABLE
2857 017730 005137 017744    COM    SUM              ;COMPLIMENT CHECKSUM
2858 017734 113714 017744    MOVB   SUM,(R4)        ;PUT COMP CHECK SUM INTO TABLE
2859 017740 005005          CLR    R5              ;CLEAR TEMP #5 - FLAG DONE MODULE
2860 017742 000207          ENLDL: RTS    PC        ;RETURN
-----
2861
2862 017744 000000          SUM:    0
2863 017746 000000          DATBYT: 0
2864 017750 000000          PAT:    0
2865
;MOD 1.3.1 ----- END MODULE -----

```

2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882 017752 005037 020634
2883 017756 005037 020642
2884 017762 005037 020640
2885 017766 112737 000177 020642
2886 017774 113737 020650 020640
2887 0200J2 005037 020646
2888 020006 113737 020652 020646
2889 020014 005037 020644
2890 020020 113737 020650 020644
2891 020026 013737 020646 020636
2892 020034 163737 020644 020636
2893 020042 005237 020636
2894 020046 002005
2895 020050 012737 100000 002274
2896 020056 000137 020632
2897 020062 013737 002202 020654
2898 020070 142737 000377 020126
2899 020076 005737 020654
2900 020102 001003
2901 020104 012737 000007 020654
2902 020112 013704 020654
2903 020116 005304
2904 020120 006304
2905 020122 150437 020126
2906 020126 000777
2907 020130 000137 020164
2908 020134 000137 020220
2909 020140 000137 020254
2910 020144 000137 020272
2911 020150 000137 020340
2912 020154 000137 020422
2913 020160 000137 020476
2914
2915 020164 123737 020646 020642
2916 020172 001004
2917 020174 012737 177777 020640
2918 020202 000405
2919 020204 113737 020644 020640
2920 020212 005237 020644
2921 020216 000565
2922
2923 020220 123737 020644 020642

.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       INCREMENT THEN DECREMENT TRACKS
4       BOUNCE BETWEEN ID AND OD
5       BOUNCE BETWEEN DECREASING ID & INCREASING OD
6       STROBE BETWEEN OD AND DECREASING ID
7       RANDOM TRACK SELECTION
-----
STKSEQ: CLR   TKTBLT      ;CLEAR TRK TBL PTR
        CLR   PRESTK     ;CLEAR PRESENT TRK
        CLR   TARGET     ;CLEAR TARGET TRK
        MOVB  #177,PRESTK ;INIT PRESENT TRK TO HANDLE TRK #0
        MOVB  OD,TARGET  ;INIT OD AS TARGET TRACK
        CLR   XID        ;INIT WORDING ID AND OD LOCATIONS
        MOVB  ID,XID     ;SAVE INSIDE DIA. IN TEMP INSIDE DIA.
        CLR   XOD        ;CLEAR TEMP OUTSIDE DIA
        MOVB  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,ERRSY ;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$:     MOV   SEQUEN,R4   ;GET SEQUENCE BITS
        DEC   R4        ;ADJUST FOR CORRECT OFFSET
        BRONTK: BR      R4,@#BRONTK ;THIS BR INST. IS MODIFIED SELECTED TRACK SEQUEN
        JMP   SEQ1      ;BRANCH TO SELECTED TRACK SEQUENCE
        JMP   SEQ2
        JMP   SEQ3
        JMP   SEQ4      ;BOUNCE ID TO OD
        JMP   SEQ5      ;DECREASING BOUNCE
        JMP   SEQ6      ;STROBE
        JMP   SEQ7      ;RANDOM
-----
SEQ1:   CMPB  XID,PRESTK  ;IF PRESENT TRACK=ID
        BNE   1$        ;THEN
        MOV   #-1,TARGET ;TERMINATE TABLE
        BR   2$        ;END SEQ1
1$:     MOVB  XOD,TARGET  ;ELSE SET NEW TRACK-OUTSIDE DIA
        INC   XOD       ;INCREMENT OUTSIDE DIA
2$:     BR   NEWTRK     ;END SEQ1
-----
SEQ2:   CMPB  XOD,PRESTK ;

```



```

2924 020226 0C1004          BNE      1$
2925 020230 012737 177777 020640  MOV     #-1,TARGET      ;TERMINATE TABLE
2926 020236 000405          BR      2$              ;END SEQ2
2927 020240 013737 020646 020640 1$:    MOV     XID,TARGET     ;SET NEXT TRACK=INSIDIA
2928 020246 005337 020646          DEC     XID              ;DECREMENT INSIDE DIA
2929 020252 000547          2$:    BR      NEWTRK
-----
2930
2931 020254 005701          SEQ3:  TST     R1              ;IF MODE
2932 020256 001402          BEQ     1$              ;NOT EQUAL TO ZERO
2933 020260 005001          CLR     R1              ;THEN CHANGE MODE
2934 020262 000756          BR      SEQ2            ;DO SEQ2
2935 020264 012701 000001          1$:    MOV     #1,R1        ;ELSE CHANGE MODE
2936 020270 000735          BR      SEQ1            ;DO SEQ1
-----
2937
2938 020272 005701          SEQ4:  TST     R1              ;IF MODE
2939 020274 001405          BEQ     1$              ;NOT EQUAL TO ZERO
2940 020276 113737 020644 020640  MOVVB  XOD,TARGET     ;THEN SET NEXT TRACK=OUTSIDE DIA
2941 020304 005001          CLR     R1              ;CHANGE MODE
2942 020306 000405          BR      2$              ;BR
2943 020310 113737 020646 020640 1$:    MOVVB  XID,TARGET     ;ELSE SET NEXT TRACK=INSIDE DIA
2944 020316 012701 000001          MOV     #1,R1          ;TERMINATE TABLE
2945 020322 005337 020636          2$:    DEC     TRKNT
2946 020326 001003          BNE     3$
2947 020330 012737 177777 020640  MOV     #-1,TARGET     ;TERMINATE TABLE
2948 020336 000515          3$:    BR      NEWTRK
-----
2949
2950 020340 123737 020646 020644  SEQ5:  CMPB   XID,XOD        ;IF INSIDE & OUTSIDE DIA
2951 020346 001421          BEQ     2$              ;NOT EQUAL
2952 020350 005701          TST     R1              ;THEN, IF MODE
2953 020352 001407          BEQ     1$
2954 020354 005001          CLR     R1              ;CHANGE MODE
2955 020356 013737 020644 020640  MOV     XOD,TARGET     ;SET NEXT TRACK=OUTSIDE DIA
2956 020364 005237 020644          INC     XOD              ;INCREMENT OUTSIDE DIA
2957 020370 000413          BR      3$              ;END SEQ5
2958 020372 012701 000001          1$:    MOV     #1,R1        ;CHANGE MODE
2959 020376 013737 020646 020640  MOV     XID,TARGET     ;SET NEXT TRACK=INSIDE DIA
2960 020404 005337 020646          DEC     XID              ;DECREMENT INSIDE DIA
2961 020410 000403          BR      3$              ;END SET5
2962 020412 012737 177777 020640  2$:    MOV     #-1,TARGET  ;TERMINATE TABLE
2963 020420 000464          3$:    BR      NEWTRK
-----
2964
2965 020422 123737 020646 020644  SEQ6:  CMPB   XID,XOD        ;IF O.D. JUST DONE
2966 020430 001416          BEQ     1$
2967 020432 123737 020642 020644  CMPB   PRESTK,XOD     ;THEN
2968 020440 001006          BNE     3$              ;SET TO DO I.D.
2969 020442 113737 020646 020640  MOVVB  XID,TARGET     ;DECREMENT I.D. FOR NEXT
2970 020450 005337 020646          DEC     XID
2971 020454 000407          BR      2$
2972 020456 113737 020644 020640  3$:    MOVVB  XOD,TARGET  ;ELSE SET TO DO O.D.
2973 020464 000403          BR      2$
2974 020466 012737 177777 020640  1$:    MOV     #-1,TARGET
2975 020474 000436          2$:    BR      NEWTRK
-----
2976
2977 020476 000240          SEQ7:  NOP
2978 020500 004737 004560          JSR     PC,RANGEN      ;GET A RANDOM NUMBER
2979 020504 042737 177600 004652  BIC     #177600,RANUM  ;CLEAR ALL BUT LOW 7 BITS

```

```
2980 020512 123737 004652 020646 IDCOMP: CMPB RANUM,XID ;IF RANUM LARGER THAN ID ADDRESS
2981 020520 003401 BLE ODCOMP ;THEN
2982 020522 000765 BR SEQ7 ;BR TO GET ANOTHER RANDOM NUMBER
2983 020524 123737 004652 020644 ODCOMP: CMPB RANUM,XOD ;IF RANUM SMALLER THAN OD ADDRESS
2984 020532 002001 BGE PRESCK ;THEN
2985 020534 000760 BR SEQ7 ;BR TO GET ANOTHER RANDOM NUMBER
2986 020536 123737 004652 020642 PRESCK: CMPB RANUM,PRESCK ;IF RANUM EQUALS PRESENT TRACK
2987 020544 001754 BEQ SEQ7 ;GET ANOTHER RANDOM NUMBER
2988 020546 013737 004652 020640 MOV RANUM,TARGET ;RANUM OK PUT IT IN TARGET TRACK
2989 020554 005337 020636 DEC TRKCNT
2990 020560 001003 BNE 1$
2991 020562 012737 177777 020640 MOV #-1,TARGET ;TERMINATE TABLE
2992 020570 000400 1$: BR NEWTRK
2993 -----
2994 020572 012702 033553 NEWTRK: MOV #TRKTBL-1,R2
2995 020576 005237 020634 INC TKTBPT
2996 020602 063702 020634 ADD TKTBPT,R2
2997 020606 113712 020640 MOV#B TARGET,(R2)
2998 020612 005737 020640 TST TARGET
2999 020616 100405 BMI ENDTKS
3000 020620 113737 020640 020642 MOV#B TARGET,PRESCK
3001 020626 000137 020126 JMP BRONTK
3002 020632 000207 ENDTKS: RTS PC
3003 -----
3004 020634 000000 TKTBPT: 0 ;TRACK TABLE POINTER
3005 020636 000000 TRKCNT: 0 ;TRACK COUNT
3006 020640 000000 TARGET: 0 ;TARGET TRACK
3007 020642 000000 PRESCK: 0 ;PRESENT TRACK
3008 020644 000000 XOD: 0 ;X OUTSIDE DIA.
3009 020646 000000 XID: 0 ;X INSIDE DIA.
3010 020650 000000 OD: 0 ;OUTSIDE DIA.
3011 020652 000000 ID: 0 ;INSIDE DIA.
3012 020654 000000 SEQUEN: 0 ;SEQUENCE #
3013 ;MOD 1.3.2 ----- END MODULE -----
3014
3015
3016
3017
3018 .SBTTL MOD 1.3.3 - CLEAR STATISTICAL TABLES
3019 -----
3020
3021 020656 012701 007314 CLRSTA: MOV #READSC,R1 ;SET UP BEGINNING ADDRESS
3022 020662 012702 011240 MOV #ENDST,R2 ;SET UP TABLE LENGTH
3023 020666 005021 BDA133: CLR (R1)+ ;CLEAR ADDRESSED LOCATION
3024 020670 020102 CMP R1,R2 ;
3025 020672 001375 BNE BDA133 ;DO UNTIL LAST ADDRESS DONE
3026 020674 000207 END133: RTS PC ;RETURN
3027 ;MOD 1.3.3 ----- END MODULE -----
```

```

3030
3031
3032
3033
3034
3035 020676 000240
3036 020700 005737 014016
3037 020704 001417
3038 020706 012737 000001 024252
3039 020714 005037 021430
3040 020720 005037 021432
3041 020724 005037 021434
3042 020730 005037 021442
3043 020734 005037 021444
3044 020740 005037 021452
3045 020744 005037 021446
3046 020750 033737 021426 002232
3047 020756 001406
3048 020760 004737 004756
3049 020764 013737 005024 002234
3050 020772 000410
3051 020774 006337 021426
3052 021000 022737 000020 021426
3053 021006 003360
3054 021010 000137 021404
3055 021014
3056 021016 013737 002176 021424
3057 021024 004737 021454
3058 021030 013737 022156 021422
3059 021036 032737 000400 021422
3060 021044 001514
3061 021046 004737 032352
3062 021052 032737 004000 021422
3063 021060 001001
3064 021062 000411
3065 021064 023727 021432 000003
3066 021072 001065
3067 021074 013737 021432 021444
3068 021102 005037 021432
3069 021106 013737 002234 021450
3070 021114 052737 002000 021450
3071 021122 032737 001000 021422
3072 021130 001001
3073 021132 000410
3074 021134 012737 002000 021440
3075 021142 005737 021430
3076 021146 001420
3077 021150 005037 021430
3078 021154 053737 021444 002230
3079 021162 006337 021426
3080 021166 013737 002234 021450
3081 021174 052737 002000 021450
3082 021202 005037 021432
3083 021206 000504
3084 021210 005737 002244
3085 021214 001403

.SBTTL MOD 2.0 - SCHEDULE SYSTEM EXERCISE
-----
SCSYEX: NOP
IFK20: TST INITL
      BEQ ELK20
      MOV #1,INITTK
      CLR EXHCP
      CLR BTHDRV
      CLR BDVSCD
      CLR DVDNCK
      CLR DRVDN
      CLR ERTSAV
      CLR SFERR
      CLR SUTPTR,SUT
      BIT ELA20
      BEQ ELA20
      CALL CVSTUT
      MOV UNITST,UUT
      BR BDB20
      ASL SUTPTR
      DUC20: CMP #20,SUTPTR
      BGT IFA20
      JMP EDC20
      BDB20: BGNSEG
      MOV TSTN,EXN
      CALL GETTST
      MOV TSTWD,TST
      IFB20: BIT #400,TST
      BEQ ELB20
      CALL STDVDN
      IFC20: BIT #4000,TST
      BNE IFI20
      BR EIC20
      IFI20: CMP BTHDRV,#3
      BNE IFL20
      MOV BTHDRV,DRVDN
      CLR BTHDRV
      EIC20: MOV UUT,RESTK
      BIS #2000,RESTK
      IFF20: BIT #1000,TST
      BNE ELF20
      BR EIF20
      ELF20: MOV #2000,ADVTRK
      IFG20: TST EXHCP
      BEQ IFH20
      CLR EXHCP
      EIF20: BIS DRVDN,SDD
      ASL SUTPTR
      MOV UUT,RESTK
      BIS #2000,RESTK
      CLR BTHDRV
      BR END20
      IFH20: TST DELDAT
      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

```

```
3086 021216 005037 002244          CLR    DELDAT      :CLEAR DEL DATA MODE
3087 021222 000403                BR     EI#20       :BR TO END IF 'H'
3088 021224 012737 000010 002244  ELH20: MOV    #10,DELDAT :SET DEC DATA MODE
3089 021232 005037 021444          EI#20: CLR    DRV#N   :CLEAR DRV DONE
3090 021236 012737 000001 021430  MOV    #1,EXHCP    :SET EX 1/2 COMPLETE
3091 021244 000443                BR     EIB#20     :BR TO END IF 'B'
3092 021246 032737 000003 021434  IFL20: BIT    #3,BDVSCD :IF BOTH DRV SEC DONE
3093 021254 001405                BEQ    ELL#20     :THEN
3094 021256 005037 021434          CLR    BDVSCD     :CLEAR DRV SEC DONE FLAGS
3095 021262 012737 004000 021440  MOV    #4000,ADVTRK :ALLOW TRACK ADVANCE
3096 021270 004737 022320          ELL20: CALL   GTDRV     :CALL MOD 2.2 - GET A DRIVE
3097 021274 000427                BR     EIB#20     :BR TO END IF 'B'
3098 021276 053737 021440 021436  ELB20: BIS    ADVTRK,INCTRK :SET ADV TRK (IF SET BY PREV OP)
3099 021304 013737 021422 023324  MOV    TST,DRV#TST :PASS DRIVE TEST
3100 021312 004737 022504          CALL   XDTVST     :CALL MOD 2.3. - EXECUTE DRIVE TEST
3101 021316 013737 023324 025410  MOV    DRV#TST,TSTEV :PASS DRIVE TEST FOR EVAL
3102 021324 004737 025334          CALL   EVTSTR     :CALL MOD 2.4 - EVAL. TEST RESULTS
3103 021330 013701 021422          MOV    TST,R1     :GET DRV TST
3104 021334 042701 171777          BIC    #171777,R1 :SAVE TRK BITS
3105 021340 010137 021436          MOV    R1,INCTRK  :SET TRK BITS
3106 021344 005037 021440          CLR    ADVTRK     :CLEAR ADV TRK FLAG
3107 021350 005037 014016          CLR    INITL     :CLEAR INITIALIZE FLAG
3108 021354 000240                NOP              :
3109 021356 005737 002276          IFM20: TST    ERRTY   :IF ERR TYPE
3110 021362 001402                BEQ    DUB#20     :NOT=0
3111 021364 004737 030702          CALL   OTERTP    :CALL MOD 2.5 - O/P ERR TYPE
3112 021370 005737 002274          DUB20: TST    ERRSY   :DO UNLESS SYSTEM ERROR
3113 021374 001011                BNE    END#20     :NOT=0 THEN
3114 021376                ENDSEG          :END SEGMENT FOR ERROR LOOPS
3115 021400 000137 021014          JMP    BDB#20     :BR TO END MOD
3116 021404 012737 000001 021426  EDC20: MOV    #1,SUTPTR  :SET SYS UNDER TEST PTR
3117 021412 052737 000001 021446  BIS    #1,SFERR   :SET SFT ERR
3118 021420 000207          END20: RTS    PC     :END MODULE
3119 -----
3120 021422 000000          TST:   0          :TEST FOR EXECUTION
3121 021424 000000          EXN:   0          :EXERCISE #
3122 021426 000001          SUTPTR: 1         :SYSTEM UNDER TEST POINTER
3123 021430 000000          EXHCP: 0          :EXERCISE HALF COMPLETE (EX#7) DEL DATA PASS
3124 021432 000000          BTHDRV: 0         :BOTH DRIVES DONE FLAG
3125 021434 000000          BDVSCD: 0         :BOTH DRIVE SECTORS DONE FLAG
3126 021436 000000          INCTRK: 0         :INCREMENT TRACK FLAGS
3127 021440 000000          ADVTRK: 0         :ADVANCE TRACK FLAG
3128 021442 000000          DVDNCK: 0         :DRV DONE CK FLAG
3129 021444 000000          DRV#DN: 0         :DRIVE DONE
3130 021446 000000          SFERR: 0          :SOFTWARE ERR
3131 021450 000000          RESTK: 0         :RESET TRK FLAG
3132 021452 000000          ERTSAV: 0        :ERR TYP SAVE REG
3133 -----
:MOD 2.0 ----- END MODULE -----
```

```

3136 .SBTTL MOD 2.1 - GET A TEST
3137 ;-----
3138
3139 021454 000240 GETTST: NOP ;
3140 021456 013701 021424 MOV EXN,R1 ;GET EXERCISE NUMBER
3141 021462 006301 ASL R1 ;DOUBLE EXERCISE NUMBER
3142 021464 012702 022164 MOV #EXADTB,R2 ;GET EXERCISE ADDRESS TABLE
3143 021470 060102 ADD R1,R2 ;CAL EXERCISE TO BE USED
3144 021472 011237 022154 MOV (R2),EXADR ;GET BEGIN ADR EXERCISE
3145 021476 005737 014016 IFL21: TST INITL ;IF INITIALIZE
3146 021502 001406 BEQ IFA21 ;IS SET, THEN
3147 021504 005037 022152 CLR TSTPTR ;CLEAR TST PTR
3148 021510 IFF21: INLOOP ;IF IN LOOP
3149 021512 BNCOMPLETE IFA21 ;SET, THEN
3150 021514 000137 022122 JMP EIF21 ;BR TO END IF 'F'
3151 021520 005737 002304 IFA21: TST RETRY ;IF RETRY
3152 021524 001410 BEQ IFB21 ;NOT=0, AND
3153 021526 032737 000004 002204 BIT #BIT02,SWREG ;IF RETRY ON ERROR
3154 021534 001106 BNE IFH21 ;IS NOT SET, THEN
3155 021536 032737 000004 002264 BIT #EVL,FLGDRS ;IF DRS 'EVL' FLAG
3156 021544 001102 BNE IFH21 ;IS NOT SET, THEN
3157 021546 005737 022152 IFB21: TST TSTPTR ;IF TST PTR
3158 021552 001006 BNE IFC21 ;EQUALS ZERO
3159 021554 012737 000002 022152 MOV #2,TSTPTR ;ADV. TST PTR I CMD
3160 021562 005037 022160 CLR TBPRCT ;CLEAR TABLE PAIR COUNT
3161 021566 000555 BR EIF21 ;BR TO END IF 'F'
3162 021570 005737 002262 IFC21: TST SECDN ;IF SECTOR DONE IS
3163 021574 001447 BEQ IFG21 ;SET THEN
3164 021576 005737 022160 IFK21: TST TBPRCT ;IF TABLE PAIR CNT=1,
3165 021602 001444 BEQ IFG21 ;THEN
3166 021604 062737 000002 022152 ADD #2,TSTPTR ;ADVANCE ONE TEST CMD
3167 021612 005037 022160 CLR TBPRCT ;CLEAR TABLE PAIR COUNT
3168 021616 005037 021442 CLR DVDNCK ;CLEAR DRV DONE CK FLAG
3169 021622 032737 040000 022156 IFD21: BIT #40000,TSTWD ;IF DONE CK
3170 021630 001411 BEQ IFM21 ;IS SET, THEN
3171 021632 005737 002260 TST TRKDN ;IF TRACK DONE IS
3172 021636 001406 BEQ IFM21 ;SET, THEN
3173 021640 005037 002260 CLR TRKDN ;CLEAR TRK DONE
3174 021644 012737 000001 021442 MOV #1,DVDNCK ;SET DRV DONE CK
3175 021652 000523 BR EIF21 ;BR TO END IF 'F'
3176 021654 032737 006000 022156 IFM21: BIT #6000,TSTWD ;IF ADV OR INCR TRK
3177 021662 001517 BEQ EIF21 ;IS SET, THEN
3178 021664 032737 100000 022156 IFN21: BIT #100000,TSTWD ;IF '4 CMD SEQ'
3179 021672 001404 BEQ ELN21 ;IS SET, THEN
3180 021674 162737 000010 022152 SUB #10,TSTPTR ;BACK UP 4 CMDS
3181 021702 000507 BR EIF21 ;BR TO END IF 'F'
3182 021704 162737 000004 022152 ELN21: SUB #4,TSTPTR ;BACK UP TWO TEST CMDS
3183 021712 000503 BR EIF21 ;BR TO END IF 'F'
3184 021714 005737 022160 IFG21: TST TBPRCT ;IF TABLE PAIR COUNT
3185 021720 001406 BEQ ELG21 ;EQUALS 1 THEN
3186 021722 005037 022160 CLR TBPRCT ;CLEAR TABLE PAIR COUNT
3187 021726 162737 000002 022152 SUB #2,TSTPTR ;BACK UP ONE CMD
3188 021734 000472 BR EIF21 ;BR END IF 'F'
3189 021736 005237 022160 ELG21: INC TBPRCT ;INCREMENT TABLE PAIR COUNT
3190 021742 062737 000002 022152 ADD #2,TSTPTR ;ADVANCE ONE CMD
3191 021750 000464 BR EIF21 ;BR END IF 'F'

```

```

3192 021752 032737 000010 002304      IFH21: BIT      #10,RETRY      ;IF NO DATA RETRY IS
3193 021760 001005                                BNE      IFI21      ;SET, OR
3194 021762 032737 000020 002304      BIT      #20,RETRY      ;IF NO CRC RETRY IS
3195 021770 001001                                BNE      IFI21      ;SET, THEN
3196 021772 000453                                BR       EIF21      ;BR END IF 'F'
3197 021774 032737 000002 002304      IFI21: BIT      #2,RETRY      ;IF WRITE RETRY IS
3198 022002 001412                                BEQ      IFJ21      ;SET, THEN
3199 022004 162737 000006 022152      SUB      #6,TSTPTR      ;BACK UP 3 CMDS
3200 022012 042737 000002 002304      BIC      #2,RETRY      ;CLEAR WRITE RETRY
3201 022020 012737 000003 022162      MOV      #3,TSVCT      ;SET TEST ADV COUNT=3
3202 022026 000433                                BR       EII21      ;BR TO END IF 'I'
3203 022030 032737 000004 002304      IFJ21: BIT      #4,RETRY      ;IF READ RETRY IS
3204 022036 001412                                BEQ      ELJ21      ;SET THEN
3205 022040 162737 000002 022152      SUB      #2,TSTPTR      ;BACK UP 1 CMD
3206 022046 042737 000004 002304      BIC      #4,RETRY      ;CLEAR READ RETRY
3207 022054 012737 000001 022162      MOV      #1,TSVCT      ;SET TEST ADV COUNT=1
3208 022062 000415                                BR       EII21      ;BR TO END IF 'I'
3209 022064 005337 022162      ELJ21: DEC      TSVCT      ;DECREMENT TEST ADV COUNT
3210 022070 062737 000002 022152      ADD      #2,TSTPTR      ;ADV TEST POINTER 1 CMD
3211 022076 005737 022162      IF021: TST      TSVCT      ;IF TEST ADV COUNTER
3212 022102 001007                                BNE      EIF21      ;EQUALS ZERO, THEN
3213 022104 005037 002304      CLR      RETRY      ;CLEAR RETRY
3214 022110 005237 022160      INC      TBPRCT      ;SET TABLE PAIR COUNT
3215 022114 000402                                BR       EIF21      ;BR TO END IF 'F'
3216 022116 005037 022160      EII21: CLR      TBPRCT      ;CLEAR TABLE PAIR CNT
3217 022122 013703 022152      EIF21: MOV      TSTPTR,R3      ;GET TEST POINTER
3218 022126 063703 022154      ADD      EXADR,R3      ;CAL. CUR. TEST OF THIS EXERCISE
3219 022132 011337 022156      MOV      (R3),TSTWD      ;PASS UP TEST WORD
3220 022136 105713      IFE21: TSTB     (R3)      ;IF CMD LOWER BYTE
3221 022140 002002                                BGE      EIE21      ;EQUALS -1, THEN
3222 022142 005037 022152      CLR      TSTPTR      ;RESET TEST PTR
3223 022146 000240      EIE21: NOP      ;
3224 022150 000207      RTS      PC      ;RETURN
-----
3226 022152 000000      TSTPTR: .WORD   0      ;TEST POINTER
3227 022154 000000      EXADR:  .WORD   0      ;CURRENT EXERCISE TABLE BASE ADDRESS
3228 022156 000000      TSTWD:  .WORD   0      ;TEST WORD TO PASS UP
3229 022160 000000      TBPRCT: .WORD   0      ;TABLE PAIR COUNT
3230 022162 000000      TSVCT:  .WORD   0      ;TEST ADVANCE COUNTER
3231
3232 022164 022300      EXADTB: .WORD   EX7      ;EXERCISE ADDRESS TABLE
3233 022166 022204      .WORD   EX1
3234 022170 022214      .WORD   EX2
3235 022172 022230      .WORD   EX3
3236 022174 022244      .WORD   EX4
3237 022176 022254      .WORD   EX5
3238 022200 022264      .WORD   EX6
3239 022202 022300      .WORD   EX7
-----

```

.SBTTL - EXERCISE/TEST TABLE		
3243		
3244		
3245	022204	177777
3246	022206	000000
3247	022210	044002
3248	022212	000777
3249	022214	177777
3250	022216	000000
3251	022220	000002
3252	022222	000003
3253	022224	154001
3254	022226	000777
3255	022230	177777
3256	022232	000000
3257	022234	000002
3258	022236	000003
3259	022240	174001
3260	022242	000777
3261	022244	177777
3262	022246	000003
3263	022250	064001
3264	022252	000777
3265	022254	177777
3266	022256	000003
3267	022260	044001
3268	022262	000777
3269	022264	177777
3270	022266	000000
3271	022270	000002
3272	022272	000003
3273	022274	170001
3274	022276	004777
3275	022300	177777
3276	022302	000000
3277	022304	000002
3278	022306	000003
3279	022310	172001
3280	022312	000003
3281	022314	064001
3282	022316	001777
3283		
3284		
3285		
3286		
3287		
3288		
3289		
3290		
3291		
3292		
3293		
3294		
3295		
3296		

BIT#	NUMONIC	FUNCTION
15	4CMD	4 COMMAND SEQUENCE
14	DCK	DONE CHECK
13	DATAK	DO DATA CHECK
12	RAW	READ AFTER WRITE FLAG
11	ADVTRK	ADVANCE TRACK MODE
10	INCTK	INCREMENT TRACK MODE
09	DDCHK	DEL. DATA CHECK
08	NXTUNT	GET NEXT UNIT, IF DONE LAST UNIT

;MOD 2.1 ----- END MODULE -----

3299
 3300
 3301
 3302
 3303
 3304
 3305
 3306
 3307
 3308
 3309
 3310
 3311
 3312
 3313
 3314
 3315
 3316
 3317
 3318
 3319
 3320
 3321
 3322
 3323
 3324
 3325
 3326
 3327
 3328
 3329
 3330
 3331
 3332
 3333
 3334
 3335

022320 000240
 022322 032737 000001 002234
 022330 001024
 022332 032737 000002 002234
 022340 001404
 022342 012737 000010 022502
 022350 000403
 022352 012737 000002 022502
 022360 033737 022502 002232
 022366 001404
 022370 052737 000001 002234
 022376 000427
 022400 000417
 022402 032737 000002 002234
 022410 001404
 022412 012737 000004 022502
 022420 000403
 022422 012737 000001 022502
 022430 033737 022502 002232
 022436 001404
 022440 042737 000001 002234
 022446 000403
 022450 052737 000001 002234
 022456 013704 002234
 022462 006304
 022464 010437 002240
 022470 062704 002336
 022474 011437 002334
 022500 000207
 022502 000000

.SBTTL MOD 2.2 - GET A DRIVE

```

:-----
GTDRV: NOP
IFA22: BIT #1,UUT ;IF UUT=DRIVE 0
      BNE IFD22 ;THEN
IFB22: BIT #2,UUT ;IF UNIT/SIDE UNDER TEST (UUT)
      BEQ ELB22 ;EQUALS 1
      MOV #10,TSTSUT ;SET TEST OF SYS. UNDER TEST UNIT/SIDE=1
      BR IFC22 ;BR TO IF 'C'
ELB22: MOV #2,TSTSUT ;SET TEST OF SYS. UNDER TEST UNIT/SIDE=0
IFC22: BIT TSTSUT,SUT ;IF DRIVE 1 SELECTED FOR TEST
      BEQ ELC22 ;THEN
      BIS #1,UUT ;SET UNIT UNDER TEST TO DRV #1
      BR EIE22 ;BR TO END IF 'E'
ELC22: BR THE22 ;BR TO THEN 'E'
IFD22: BIT #2,UUT ;IF UNIT/SIDE UNDER TEST (UUT)
      BEQ ELD22 ;EQUALS 1
      MOV #4,TSTSUT ;SET TEST OF SYS. UNDER TEST UNIT/SIDE 1
      BR IFE22 ;BR TO IF 'E'
ELD22: MOV #1,TSTSUT ;SET TEST OF SYS. UNDER TEST UNIT/SIDE 0
IFE22: BIT TSTSUT,SUT ;IF DRIVE 0 SELECTED FOR TEST
      BEQ ELE22 ;THEN
      BIC #1,UUT ;SET UNIT UNDER TEST TO DRV#0
      BR EIE22 ;BR TO END IF 'E'
ELE22: BIS #1,UUT ;SET UNIT UNDER TEST TO DRV#1
EIE22: MOV UUT,R4 ;GET UNIT UNDER TEST
      ASL R4 ;DOUBLE IT
      MOV R4,UUTOFF ;SET UUT OFFSET
      ADD #U00,R4 ;GET UUT UNIT # FOR PRINT
      MOV (R4),UNIT ;SET UNIT=PRINT UNIT #
END22: RTS PC ;RETURN
:-----
TSTSUT: 0
:MOD 2.2 ----- END MODULE -----

```


3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
3389
3390
3391
3392
3393

022504 013737 002252 023326
022512 013702 002240
022516 005737 002172
022522 001010
022524 032737 000002 002234
022532 001404
022534 013737 002222 002236
022542 000403
022544 013737 002220 002236
022552 005737 021450
022556 001413
022560 113705 021450
022564 006305
022566 062705 023306
022572 013715 002206
022576 005037 002262
022602 005037 021450
022606 005737 014016
022612 001415
022614 012705 023276
022620 012704 000004
022624 005025
022626 005304
022630 001375
022632 012704 000004
022636 013725 002206
022642 005304
022644 001374
022646 012701 023306
022652 060201
022654 010137 023320
022660 017737 000434 024240
022666 012701 023276
022672 060201
022674 010137 023316
022700 017737 000412 023710
022706
022710
022712 000532
022714 005737 002304
022720 001447
022722 032737 000004 002204
022730 001004
022732 032737 000004 002264
022740 001437
022742 032737 000001 002304
022750 001001
022752 000404
022754 032737 000010 002204
022762 001003
022764 005037 023322

.SBTTL MOD 2.3 - EXECUTE DRIVE TEST

```

XDVST: MOV WDCNT,WDCT ;SET DRIVE WORD CNT
MOV UUTOFF,R2 ;GET UUT OFFSET
IFA23: TST RXXX ;IF DEVICE IS AN
BNE 1$ ;RX02 THEN
BIT #2,UUT ;IF UNIT UNDER TEST IS
BEQ 1$ ;#1 THEN
MOV U1ADR,UUTADR ;GET UNIT #1 UNIBUS ADR
BR IF123 ;BR TO END IF 'A'
1$: MOV UOADR,UUTADR ;GET UNIT #0 UNIBUS ADR
IF123: TST RESTK ;IF RESET TRK
BEQ IFB23 ;IF SET, THEN
MOVB RESTK,R5 ;GET UUT OFFSET
ASL R5 ;DOUBLE OFFSET
ADD #CTRK,R5 ;ADD TRK TABLE ADR
MOV OTDITK,(R5) ;RESET TO MIN TRK
CLR SECDN ;CLEAR SEC DONE FLAG
CLR RESTK ;CLEAR RESET TRK FLAG
IFB23: TST INITL ;IF INITIALIZE IS
BEQ EIB23 ;SET, THEN
MOV #CSEC,R5 ;GET START OF CUR TRK &SEC TBL
MOV #4,R4 ;SET TBL LENGTH
1$: CLR (R5)+ ;CLEAR TABLES
DEC R4 ;DECR TBL LENGTH
BNE 1$ ;DO UNTIL LENGHT=0
MOV #4,R4 ;SET TBL LENGTH
2$: MOV OTDITK,(R5)+ ;SET STARTING TRACKS
DEC R4 ;DECREMENT TBL LENGTH
BNE 2$ ;DO UNTIL LENGTH=0
EIB23: MOV #CTRK,R1 ;GET BEGIN ADR DRIVE CURRENT TRK.
ADD R2,R1 ;CAL. DRIVE CUR. TRK. LOCATOR
MOV R1,CNTKLC ;SAVE DRV. CUR. TRK.
MOV @CNTKLC,CURTRK ;GET DRIVE CUR. TRK.
MOV #CSEC,R1 ;GET BEGIN ADR DRIVE CUR. SEC.
ADD R2,R1 ;CAL. DRIVE CUR. SEC. LOCATOR
MOV R1,CNSCLC ;SAVE DRV CUR SEC LOC.
MOV @CNSCLC,CURSEC ;GET DRIVE CUR SEC.
IFJ23: INLOOP ;IF IN LOOP
BNCOMPLETE IFC23 ;THEN
BR EIJ23 ;BR TO END IF 'I'
IFC23: TST RETRY ;IF RETRY IS
BEQ IFG23 ;NOT=0, AND
BIT #BIT02,SWREG ;IF RETRY ON ERR
BNE IFD23 ;SET OR
BIT #EVL,FLGDRS ;DRS 'EVL' FLAG
BEQ IFG23 ;IS SET, THEN
IFD23: BIT #1,RETRY ;IF SEEK RETRY
BNE 1$ ;IS = 0
BR 2$ ;THEN BR TO 2$
1$: BIT #BIT03,SWREG ;ELSE IF RECAL SWITCH
BNE THD23 ;IS NOT SET
2$: CLR SEEK ;THEN CLEAR SEEK FUNCTION FLAG

```

3394	022770	000420				BR	EID23	:BR TO END IF 'D'
3395	022772	012737	040000	024404	THD23:	MOV	#40000,DVTST	:PASS PROGRAM INITIALIZE TO DRIVE TEST
3396	023000	004737	024254			CALL	GTDVFN	:CALL MOD 2.3.3 GET DRIVE FUNCTION
3397	023004	013737	023332	023330		MOV	DRVFN,WDOT	:PASS DRIVE FUNCTION
3398	023012	013737	002236	025034		MOV	UUTADR,CSADR	:SET ADR FOR DRIVE FUNCTION
3399	023020	004737	024406			CALL	OTDVFN	:CALL MOD 2.3.4 O/P DRIVE FUNCTION
3400	023024	012737	000001	023322		MOV	#1,SEEK	:SET SEEK FLAG
3401	023032	005037	002304		EID23:	CLR	RETRY	:CLEAR RETRY FLAGS
3402	023036	000460				BR	EIJ23	:BR TO END IF 'C'
3403	023040	013705	023324		IFG23:	MOV	DRVIST,R5	:SETUP DRIVE TST
3404	023044	042705	177770			BIC	#177770,R5	:FOR TYPE CK
3405	023050	005705				TST	R5	:IF DRIVE TST
3406	023052	001404				BEQ	IFE23	:IS NOT 'FILL BUFF'
3407	023054	022705	000003			CMP	#3,R5	:OR
3408	023060	001401				BEQ	IFE23	:NOT 'READ SEC' , THEN
3409	023062	000434				BR	IFH23	:BR TO IF 'H'
3410	023064	005737	002262		IFE23:	TST	SECDN	:IF SEC DONE
3411	023070	001417				BEQ	ELE23	:IS = 1
3412	023072	005737	021436		IFF23:	TST	INCRTRK	:IF INCR TRK FLAGS
3413	023076	001414				BEQ	ELE23	:ARE SET ,THEN
3414	023100	013737	021436	024236		MOV	INCRTRK,TRKINC	:PASS TRK FLAGS
3415	023106	004737	023742			CALL	GETTRK	:CALL MOD 2.3.2 GET TRACK
3416	023112	013777	024240	000200		MOV	CURTRK,@CNTKLC	:SAVE CURRENT TRACK
3417	023120	012737	000001	023322		MOV	#1,SEEK	:SET SEEK FLAG
3418	023126	000402				BR	EIE23	:BR TO END IF 'E'
3419	023130	005037	023322		ELE23:	CLR	SEEK	:RESET SEEK
3420	023134	017737	000156	023710	EIE23:	MOV	@CNSCLC,CURSEC	:PASS CURRENT SECTOR
3421	023142	004737	023334			CALL	GETSEC	:CALL MOD 2.3.1 GET A SECTOR
3422	023146	013777	023710	000142		MOV	CURSEC,@CNSCLC	:SAVE UPDATED CURRENT SECTOR
3423	023154	032737	000006	023324	IFH23:	BIT	#6,DRVIST	:IF DRIVE TST
3424	023162	001006				BNE	EIJ23	:IS 'FILL BUFF' ,THEN
3425	023164	012701	034006			MOV	#DATPAT,R1	:SET UP DATA PATTERN ADR
3426	023170	117721	000124			MOVB	@CNTKLC,(R1)+	:SET TRK ADR IN DATA 3UF BYTE #0
3427	023174	117711	000116			MOVB	@CNSCLC,(R1)	:SET SEC ADR IN DATA BUF BYTE#1
3428	023200	005037	024404		EIJ23:	CLR	DVTST	:CLEAR DRIVE TEST
3429	023204	113737	023324	024404		MOVB	DRVIST,DVTST	:PASS DRIVE TEST
3430	023212	004737	024254			CALL	GTDVFN	:CALL MOD 2.3.3 GET DRIVE FUNCTION
3431	023216	013737	023332	002332		MOV	DRVFN,CMD	:SET COMMAND FOR PRINT
3432	023224	013737	023332	023330		MOV	DRVFN,WDOT	:PASS FUNCTION WORD (PASS TO 2.3.4)
3433	023232	017737	000062	025036		MOV	@CNTKLC,TRKADR	:PASS CURRENT TRACK (PASS TO 2.3.4)
3434	023240	017737	000052	025040		MOV	@CNSCLC,SECADR	:PASS CURRENT SECTOR (PASS TO 2.3.4)
3435	023246	013737	002236	025034		MOV	UUTADR,CSADR	:PASS UUT C&S ADR (PASS TO 2.3.4)
3436	023254	004737	024406			CALL	OTDVFN	:CALL MOD 2.3.4 O/P DRIVE FUNCTION
3437	023260	013737	025036	002254		MOV	TRKADR,TRACK	:SAVE TRACK ADDR IN GLOBAL
3438	023266	013737	025040	002256		MOV	SECADR,SECTOR	:SAVE SECTOR ADDR IN GLOBAL
3439	023274	000207				RTS	PC	:RETURN
3440								

```
3443  
3444  
3445 023276 000000  
3446 023300 000000  
3447 023302 000000  
3448 023304 000000  
3449 023306 000000  
3450 023310 000000  
3451 023312 000000  
3452 023314 000000  
3453  
3454 023316 000000  
3455 023320 000000  
3456 023322 000000  
3457 023324 000000  
3458 023326 000000  
3459 023330 000000  
3460 023332 000000  
3461
```

```
-----  
CSEC: .WORD 0 ;CURRENT DRV SECTOR TABLE  
      .WORD 0  
      .WORD 0  
      .WORD 0  
CTRK: .WORD 0 ;CURRENT DRV TRK TABLE  
      .WORD 0  
      .WORD 0  
      .WORD 0  
CNSCLC: .WORD 0 ;CURRENT SECTOR LOCATOR  
CNTKLC: .WORD 0 ;CURRENT TRACK LOCATOR  
SEEK: .WORD 0 ;SEEK FLAG  
DRVTST: .WORD 0 ;DRIVE TEST  
WDCT: .WORD 0 ;WORD COUNT  
WDOT: .WORD 0 ;FUNCTION WORD TO SEND OUT  
DRVFN: .WORD 0 ;DRIVE FUNCTION WORD  
;MOD 2.3 ----- END MODULE -----
```

```

3464      .SBTTL  MOD 2.3.1 - GET A SECTOR
3465      :-----
3466
3467 023334 005037 023704      GETSEC: CLR      UTSCDN      ;CLEAR UUT SECTOR DONE
3468 023340 013705 002234      MOV      UUT,R5      ;GET UNIT UNDER TST
3469 023344 006305              ASL      R5          ;DOUBLE FOR WRD ADR
3470 023346 005737 014016      IFI231: TST      INITL      ;IF INITIALIZE IS
3471 023352 001406              BEQ      EI1231      ;SET, THEN
3472 023354 012701 023664      MOV      #SSEC,R1    ;GET STARTING SEC ADR
3473 023360 005021              CLR      (R1)+      ;CLEAR UNT00 SSEC
3474 023362 005021              CLR      (R1)+      ;CLEAR UNT01 SSEC
3475 023364 005021              CLR      (R1)+      ;CLEAR UNT10 SSEC
3476 023366 005011              CLR      (R1)       ;CLEAR UNT11 SSEC
3477 023370 012701 023664      EI1231: MOV      #SSEC,R1 ;GET START SECTOR BASE ADR
3478 023374 060501              ADD      R5,R1      ;FIND ADR UUT START SECTOR (TEMP 1)
3479 023376 011102              MOV      (R1),R2    ;SAVE UUT STARTING SECTOR (TEMP 2)
3480 023400 012703 023674      MOV      #NSEC,R3    ;GET NEXT SECTOR BASE ADR
3481 023404 060503              ADD      R5,R3      ;FIND ADR UUT NEXT SECTOR (TEMP 3)
3482 023406 011304              MOV      (R3),R4    ;SAVE UUT NEXT SECTOR (TEMP 4)
3483 023410 020237 002212      IFA231: CMP      R2,MINSEC ;IF STARTING SECTOR < MIN. SECTOR
3484 023414 103422              BLO      ELA231      ;THEN
3485 023416 010437 023710      MOV      R4,CURSEC   ;SET CURRENT SECTOR=UUT NEXT SECTOR
3486 023422 023737 023660 023706  IFG231: CMP      SCPSCT,INTLV ;IF SECTOR PASS CNT< INTERLV
3487 023430 103053              BHIS     THF231      ;THEN BR TO THEN 'F',ELSE
3488 023432 005737 023662      IFH231: TST      STSCFG   ;IF START SEC FLAG
3489 023436 001405              BEQ      ELH231      ;IS SET, THEN
3490 023440 005037 023662      CLR      STSCFG     ;CLEAR FLAG
3491 023444 010204              MOV      R2,R4      ;SET DRV NXT SEC= DRV START SEC
3492 023446 010213              MOV      R2,(R3)    ;SAVE DRV NXT SEC
3493 023450 000426              BR       IFC231      ;BR TO IF 'C'
3494 023452 063704 023706      ELH231: ADD      INTLV,R4 ;NSEC=NSEC+INTERLV
3495 023456 010413              MOV      R4,(R3)    ;SAVE NEXT SEC
3496 023460 000422              BR       IFC231      ;BR TO IF 'C'
3497 023462 013737 002212 023710  ELA231: MOV      MINSEC,CURSEC ;SET CURRENT SECTOR = MIN. SECTOR
3498 023470 013711 002212      MOV      MINSEC,(R1) ;SET UUT START SECTOR = MIN. SECTOR
3499 023474 013702 002212      MOV      MINSEC,R2  ;SET R2=MINSEC
3500 023500 005037 023660      CLR      SCPSCT     ;CLEAR SECTOR PASS COUNT
3501 023504 023737 002212 002214  IFB231: CMP      MINSEC,MAXSEC ;IF MAX. SECTOR NOT=MIN. SECTOR
3502 023512 001443              BEQ      ELB231      ;THEN
3503 023514 010205              THB231: MOV      R2,R5 ;GET UUT STARTING SECTOR
3504 023516 063705 023706      ADD      INTLV,R5   ;ADD SECTOR INTERLEAVE
3505 023522 010513              MOV      R5,(R3)    ;SAVE NEXT UUT NEXT SEC (TEMP 5)
3506 023524 010504              MOV      R5,R4      ;SAVE NEXT UUT NEXT SEC (TEMP 4)
3507 023526 020437 002214      IFC231: CMP      R4,MAXSEC ;IF NEXT SECTOR > MAX. SECTOR
3508 023532 103432              BLO      ELC231      ;THEN
3509 023534 005211              INC      (R1)       ;INCREMENT UUT STARTING SECTOR
3510 023536 011102              MOV      (R1),R2    ;SET UP NEW START SEC
3511 023540 005237 023660      INC      SCPSCT     ;INCR SECTOR PASS CNT
3512 023544 020437 002214      IFD231: CMP      R4,MAXSEC ;IF NXT SEC NOT = MAX SEC
3513 023550 001417              BEQ      ELD231      ;THEN
3514 023552 020237 002214      IFF231: CMP      R2,MAXSEC ;IF DRV START SEC > MAX SEC
3515 023556 101411              BLOS     ELF231      ;THEN
3516 023560 012737 000001 023704  THF231: MOV      #1,UTSCDN ;SET UUT SECTOR DONE
3517 023566 004737 023712      CALL     STSCDN     ;CALL MOD 2.3.1.A - SET DRIVE SECTOR DONE FLAG
3518 023572 005011              CLR      (R1)       ;CLEAR UUT STARTING SECTOR
3519 023574 005037 023660      CLR      SCPSCT     ;CLEAR SEC PASS CNT

```



```

3565 .SBTTL MOD 2.3.2 - GET A TRACK
3566 :-----
3567
3568 023742 013737 002210 024234 GETTRK: MOV INDITK,MAXTRK :GET INSIDE DIA AS SET BY OP
3569 023750 013737 002206 024232 MOV OTDITK,MINTRK :GET OUTSIDE DIA AS SET BY OP
3570 023756 005737 024252 IFH232: TST INITK :IF INITIALIZE TRK IS
3571 023762 001413 BEQ EIH232 :SET, THEN
3572 023764 005037 024252 CLR INITK :RESET INITIALIZE TRK FLG
3573 023770 012701 024242 MOV #TKTL,R1 :GET START OF TRK TBL
3574 023774 005021 CLR (R1)+ :SET UNT00
3575 023776 005021 CLR (R1)+ :SET UNT01
3576 024000 005021 CLR (R1)+ :SET UNT10
3577 024002 005011 CLR (R1) :SET UNT11
3578 024004 013757 024232 024240 MOV MINTRK,CURTRK :SET MIN CURRENT TRK
3579 024012 013702 002234 EIH232: MOV UUT,R2 :GET UNIT UNDER TEST INDICATOR
3580 024016 005302 ASL R2 :DOUBLE FOR ADDRESSING WORDS
3581 024020 005037 024230 CLR TRKDNF :CLEAR TRACK DONE FLAG
3582 024024 032737 002000 024236 IFA232: BIT #2000,TRKINC :IF INCREMENT TRACK FLAG
3583 024032 001023 BNE IFG232 :NOT SET, THEN (USE SELECTED TRK SEQ)
3584 024034 012701 024242 MOV #TKTL,R1 :GET DRIVE TRACK TABLE LOCATOR BASE ADR
3585 024040 060201 ADD R2,R1 :CAL. DRV. TRK. TAB. LOCATOR ADR
3586 024042 011102 MOV (R1),R2 :GET DRV. TRK. TAB. LOCATOR
3587 024044 012703 033554 MOV #TRKTBL,R3 :GET BEGIN TRACK TABLE ADR
3588 024050 060203 ADD R2,R3 :CAL. TRACK TAB. ADR. THIS DRIVE
3589 024052 005202 INC R2 :INCREMENT DRV. TRK. TAB. LOCATOR
3590 024054 010211 MOV R2,(R1) :SAVE DRV. TRK. TAB. LOCATOR
3591 024056 111337 024240 MOV#B (R3),CURTRK :SAVE CURRENT TRACK
3592 024062 005203 INC R3 :INCREMENT TRACK TAB. POINTER
3593 024064 105713 IFF232: TSTB (R3) :IF NEXT TRACK
3594 024066 002004 BGE ELF232 :EQUALS -1
3595 024070 012737 000001 024230 MOV #1,TRKDNF :THEN SET TRACK DONE FLAG
3596 024076 005011 CLR (R1) :RESET DRV. TRK. TAB. LOCATOR ADR.
3597 024100 000445 ELF232: BR END232 :BR TO END MOD.
3598 024102 123737 024240 024234 IFG232: CMPB CURTRK,MAXTRK :IF CURRENT TRK > OR = MAX TRK (O. D.)
3599 024110 103403 BLO IFB232 :THEN
3600 024112 013737 024232 024240 MOV MINTRK,CURTRK :SET CURRENT TRK = MIN TRK
3601 024120 123737 024240 024232 IFB232: CMPB CURTRK,MINTRK :IF CURRENT TRK > OR = MIN TRK (O.D.)
3602 024126 103427 BLO ELB232 :THEN
3603 024130 013701 024240 MOV CURTRK,R1 :GET CURRENT TRACK
3604 024134 005201 INC R1 :INCREMENT CURRENT TRACK
3605 024136 120137 024234 IFC232: CMPB R1,MAXTRK :IF CURRENT TRK +1 < MAX TRK (I.D.)
3606 024142 103001 BHIS IFD232 :THEN
3607 024144 000406 BR EID232 :BRANCH TO END IF 'D'
3608 024146 120137 024234 IFD232: CMPB R1,MAXTRK :IF CURRENT TRK +1 = MAX TRK
3609 024152 001006 BNE IFE232 :THEN
3610 024154 012737 000001 024230 MOV #1,TRKDNF :SET TRK DONE FLAG
3611 024162 010137 024240 EID232: MOV R1,CURTRK :SAVE CURRENT TRK +1 = CURRENT TRK
3612 024166 000412 BR END232 :BR END OF MOD.
3613 024170 123737 024234 024232 IFE232: CMPB MAXTRK,MINTRK :IF TRK MAX = TRK MIN
3614 024176 001003 BNE ELB232 :THEN
3615 024200 012737 000001 024230 MOV #1,TRKDNF :SET TRK DONE FLAG
3616 024206 013737 024232 024240 ELB232: MOV MINTRK,CURTRK :SET CURRENT TRK = MIN. TRK (O.D.)
3617 024214 013737 024230 002260 END232: MOV TRKDNF,TRKDN :SAVE TRACK DONE FLAG
3618 024222 005037 024236 CLR TRKINC :CLEAR TRK INCR FLAG
3619 024226 000207 RTS PC :
3620 :-----

```

```
3623  
3624 024230 000000  
3625 024232 000000  
3626 024234 000000  
3627 024236 000000  
3628 024240 000000  
3629 024242 000000  
3630 024244 000000  
3631 024246 000000  
3632 024250 000000  
3633 024252 000000  
3634
```

```
-----  
;TRKDNF: .WORD 0 ;TRACK DONE FLAG  
;MINTRK: .WORD 0 ;MINIMUM TRACK - O.D.  
;MAXTRK: .WORD 0 ;MAXIMUM TRACK - I.D.  
;TRKINC: .WORD 0 ;INCREMENT TRK FLAG  
;CURTRK: .WORD 0 ;CURRENT TRACK  
;TKTL: .WORD 0 ;DRV TRK TABLE LOCATOR  
;INITTK: .WORD 0 ;INITIALIZE TRK FLAG  
;MOD 2.3.2 ----- END MODULE -----
```

```
3637 .SBTTL MOD 2.3.3 - GET A DRIVE FUNCTION
3638 ;-----
3639
3640 GTDVFN: CLR R1 ;CLEAR REG #1
3641 024254 005001 024404 MOV DVTST,R1 ;GET DRIVE TEST
3642 024256 013701 040000 IFA233: BIT #40000,R1 ;IF NOT INITIALIZE
3643 024262 032701 040000 BNE IFB233 ;THEN
3644 024266 001012 177700 BIC #177700,R1 ;CLEAR TOP BYTE OF R1
3645 024270 042701 177700 ASL R1 ;FORMAT FUNCTION
3646 024274 006301 000001 BIS #1,R1 ;SET GO BIT
3647 024276 052701 000001 IFE233: CMP R1,#5 ;IF WRT FUNCT
3648 024302 020127 000005 BNE IFB233 ;THEN
3649 024310 053701 002244 BIS DELDAT,R1 ;SET DEL DAT WRT (IF SET)
3650 024314 005737 002172 IFB233: TST RXXX ;IF DRIVE IS RXXX
3651 024320 001411 BEQ IFD233 ;THEN
3652 024322 032737 000002 002234 IFC233: BIT #2,UUT ;IF SIDE #1 IS SELECTED
3653 024330 001403 BEQ ELC233 ;THEN
3654 024332 052701 001000 BIS #1000,R1 ;SET SIDE #1 BIT
3655 024336 000402 BR IFD233 ;BRANCH TO IF 'D'
3656 024340 042701 001000 ELC233: BIC #1000,R1 ;SET FOR SIDE #0
3657 024344 032737 000001 002234 IFD233: BIT #1,UUT ;IF UNIT UNDER TEST IS
3658 024352 001403 BEQ ELD233 ;DRIVE #1
3659 024354 052701 000020 BIS #20,R1 ;THEN SET DRIVE #1 SELECT BIT
3660 024360 000402 BR EID233 ;BRANCH TO IF 'D'
3661 024362 042701 000020 ELD233: BIC #20,R1 ;ELSE CLEAR DRIVE #1 SELECT BIT
3662 024366 053701 002242 EID233: BIS DEN,R1 ;SET DENSITY BIT
3663 024372 052701 000100 BIS #100,R1 ;SET INTERRUPT BIT
3664 024376 010137 023332 MOV R1,DRVFN ;PASS UP FUNCTION WORD
3665 024402 000207 END233: RTS PC ;RETURN
3666 ;-----
3667 024404 000000 DVTST: 0 ;DRIVE TEST WORD
3668 ;MOD 2.3.3 ----- END MODULE -----
```



```

3671
3672
3673
3674 024406 013701 025034
3675 024412 062701 000002
3676 024416 010137 025032
3677 024422 012737 000040 025330
3678 024430 013737 023330 025022
3679 024436 013737 025034 025024
3680 024444 004737 025042
3681 024450 032737 040000 023330
3682 024456 001402
3683 024460 000137 025016
3684 024464 032737 000010 023330
3685 024472 001043
3686 024474 032737 000004 023330
3687 024502 001047
3688 024504 012737 000200 025330
3689 024512 013737 023326 025022
3690 024520 013737 025032 025024
3691 024526 004737 025042
3692 024532 032737 000002 023330
3693 024540 001004
3694 024542 012737 034006 025022
3695 024550 000403
3696 024552 012737 034406 025022
3697 024560 012737 000200 025330
3698 024566 013737 025032 025024
3699 024574 004737 025042
3700 024600 000444
3701 024602 032737 000004 023330
3702 024610 001455
3703 024612 032737 000002 023330
3704 024620 001035
3705 024622 012737 000200 025330
3706 024630 013737 025040 025022
3707 024636 042737 177700 025022
3708 024644 013737 025032 025024
3709 024652 004737 025042
3710 024656 013737 025036 025022
3711 024664 042737 177600 025022
3712 024672 012737 000200 025330
3713 024700 013737 025032 025024
3714 024706 004737 025042
3715 024712 000437
3716 024714 012737 000200 025330
3717 024722 012737 033544 025022
3718 024730 013737 025032 025024
3719 024736 004737 025042
3720 024742 000423
3721 024744 032737 000002 023330
3722 024752 001404
3723 024754 012737 000001 025026
3724 024762 000413
3725 024764 012737 000200 025330
3726 024772 013737 025030 025022

```

.SBTTL MOD 2.3.4 - OUTPUT DRIVE FUNCTION

```

-----
OTDVFN: MOV CSADR,R1 ;GET STATUS REG ADR
ADD #2,R1 ;ADD 2 TO ADR
MOV R1,DBADR ;SAVE AS DATA ADDRESS
MOV #DNBIT,RDYWD ;READY TEST WD (PASS TO 2.3.4.1)
MOV WDOT,WRDS ;WORD FOR OUTPUT (PASS TO 2.3.4.1)
MOV CSADR,ADRS ;ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
JSR PC,OUTSWD ;OUTPUT FUNCTION WD (FW) DO 2.3.4.1)
IFA234: BIT #40000,WDOT ;IF FUNCTION IS
BEQ THA234 ;NOT AN "INITIALIZE" (FW BIT#14=0)
JMP END234 ;THEN,
THA234: BIT #10,WDOT ;IF FUNCTION IS
BNE IFC234 ;"READ, WRITE, FILL, EMPTY" (FW BIT#3=0)
IFH234: BIT #4,WDOT ;AND THEN IF FUNCTION IS
BNE ELH234 ;"EMPTY, FILL" (FW BIT#2=0)
MOV #TRBIT,RDYWD ;THEN SET OUTPUT READY TEST WORD (PASS TO 2.3.4.1)
MOV WDCT,WRDS ;AND SET WORD FOR OUTPUT (PASS TO 2.3.4.1)
MOV DBADR,ADRS ;AND SET ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
JSR PC,OUTSWD ;OUTPUT BASE ADDRESS WORD DO 2.3.4.1)
IFK234: BIT #2,WDOT ;IF "FILL" (FW BIT#1=0)
BNE ELK234 ;THEN
MOV #DATPAT,WRDS ;SET DATA PATTERN ADR (PASS TO 2.3.4.1)
BR EIK234 ;BR TO END IF "K"
ELK234: MOV #DATBUF,WRDS ;SET DATA BUFFER ADR (PASS TO 2.3.4.1)
EIK234: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD (PASS TO 2.3.4.1)
MOV DBADR,ADRS ;ADDRESS OF OUTPUT (PASS TO 2.3.4.1)
JSR PC,OUTSWD ;OUTPUT WORD COUNT WORD DO 2.3.4.1)
BR EIH234 ;BRANCH TO END IF "H"
IFC234: BIT #4,WDOT ;IF FUNCTION WORD IS
BEQ IFE234 ;"WRITE D.D" OR "READ E.C" (FW BIT #2=1)
IFD234: BIT #2,WDOT ;THEN, IF FUNCTION IS
BNE ELD234 ;"WRITE D.D", THEN (FW BIT#1=0)
ELH234: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
MOV SECADR,WRDS ;MOVE TRACK AND SECTOR ADDRESS
BIC #177700,WRDS ;FORMAT TO SECTOR ADDRESS
MOV DBADR,ADRS ;ADDRESS OF OUTPUT
JSR PC,OUTSWD ;OUTPUT SECTOR ADDRESS
MOV TRKADR,WRDS ;MOVE TRACK AND SECTOR ADDRESS
BIC #177600,WRDS ;FORMAT TRACK ADDRESS
MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WORD
MOV DBADR,ADRS ;ADDRESS OF OUTPUT
JSR PC,OUTSWD ;OUTPUT TRACK ADDRESS
EIH234: BR EIB234 ;ENDIF H -DONE
ELD234: MOV #TRBIT,RDYWD ;SET READY WD TO TR MODE
MOV #XERUUT,WRDS ;EXT ERR. CODE TABLE ADD
MOV DBADR,ADRS ;ADDRESS OF OUTPUT, RXDB
JSR PC,OUTSWD ;O/P BASE ADD FOR ERR. CODE
BR EIB234 ;DONE
IFE234: BIT #2,WDOT ;IF FUNCTION IS
BEQ ELE234 ;"READ STATUS" (FW BIT#1=1)
THE234: MOV #1,ERSTAT ;THEN-SET ERR STATUS FLAG
BR EIB234 ;DONE
ELE234: MOV #TRBIT,RDYWD ;SET OUTPUT READY TEST WD
MOV VALWD,WRDS ;VALIDATION WORD

```

```

3727 025000 013737 025032 025024
3728 025006 004737 025042
3729 025012 004737 025104
3730 025016 000240
3731 025020 000207
3732
3733 025022 000000
3734 025024 000000
3735 025026 000000
3736 025030 000111
3737 025032 000000
3738 025034 000000
3739 025036 000000
3740 025040 000000
3741
3742
3743
3744
3745
3746
3747
3748
3749 025042 000240
3750 025044 013737 025034 025332
3751 025052 013737 025330 025330
3752 025060 004737 025230
3753 025064 033777 025330 177742
3754 025072 001403
3755 025074 013777 025022 177722
3756 025102 000207
3757

```

```

MOV DBADR,ADRS ;ADDRESS OF OUTPUT, RXDB
JSR PC,OUTSWD ;OUTPUT VALIDATION WORD
EIB234: CALL WATCH ;CALL MOD U.2 -WATCH DOG
END234: NOP ;
RTS PC ;RETURN TO MOD 2.3
-----
WRDS: 0 ;MODULE 2.3.4.1 OUTPUT WORD
ADRS: 0 ;MODULE 2.3.4.1 OUTPUT ADDRESS
ERSTAT: 0 ;MODULE 0.0 ERR STATUS READ FLAG
VALWD: 111 ;EXTERNAL, VALIDATION WD (SET DENS-ASCII 'I')
DBADR: 0 ;RX DATA BUFFER ADDRESS
CSADR: 0 ;RX CONT/STATUS ADDRESS
TRKADR: 0 ;TRACK ADDRESS
SECADR: 0 ;SECTOR ADDRESS
:MOD 2.3.4 ----- END MODULE -----

```

.SBTTL MOD 2.3.4.1 - OUTPUT SINGLE WORD

```

-----
OUTSWD: NOP ;
MOV CSADR,CSRADR ;SET C&S REG ADR
MOV RDYWD,RDYWD ;OUTPUT READY WORD (PASS TO DELAY)
JSR PC,DELAY ;DELAY FOR READY DO DELAY
BIT RDYWD,@CSADR ;IF READY,
BEQ ED2341 ;THEN
MOV WRDS,@ADRS ;MOV WORD TO ADDRESS
ED2341: RTS PC ;RETURN TO MOD 2.3.4
:MOD 2.3.4.1 ----- END MODULE -----

```

```
3760 .SBTTL MOD U.2.3.4 - WATCH DOG TIMER
3761 ;-----
3762
3763 025104 005037 025226 WATCH: CLR DNFLAG ;CLEAR DONE FLAG
3764 025110 SETPRI #PRI00 ;SET PROCESSOR PRI=0 - ALLOW INTERRUPTS
3765 025116 013704 025222 MOV DX,R4 ;SET DELAY MULT
3766 025122 013703 025224 BAU234: MOV DLY,R3 ;SET DELAY
3767 025126 005737 025226 IBU234: TST DNFLAG ;IF INTERRUPTS DONE FLAG
3768 025132 001410 BEQ LBU234 ;IS SET, THEN
3769 025134 032777 000040 177672 ICU234: BIT #DNBIT,@CSADR ;IF DONT BIT
3770 025142 001023 BNE XU234 ;IS NOT SET, THEN
3771 025144 012737 010000 002276 MOV #BIT12,ERRTY ;SET INTERR, BUT NO DONE ERROR
3772 025152 000417 BR XU234 ;BR TO MOD 'EXIT'
3773 025154 005303 LBU234: DEC R3 ;DECREMENT DELAY COUNT
3774 025156 001363 UDU234: BNE IBU234 ;DO UNIT DELAY COUNT=0
3775 025160 005304 DEC R4 ;DECREMENT DELAY MULT
3776 025162 001357 UAU234: BNE BAU234 ;DO UNTIL DELAY MULT=0
3777 025164 032777 000040 177642 IEU234: BIT #DNBIT,@CSADR ;IF DONE BIT IS
3778 025172 001404 BEQ LEU234 ;SET, THEN
3779 025174 052737 020000 002276 BIS #BIT13,ERRTY ;SET DONE, BUT NO INTERRUPT ERROR
3780 025202 000403 BR XU234 ;BR TO MOD 'EXIT'
3781 025204 052737 040000 002274 LEU234: BIS #BIT14,ERRSY ;SET T.O. ERROR
3782 025212 XU234: SETPRI #PRI07 ;SET PROCESSOR PRI=7 - NO INTERRUPTS
3783 025220 000207 RTS PC ;RETURN TO MOD 2.3.4
3784 ;-----
3785 025222 000040 DX: 40 ;DELAY MULT
3786 025224 100000 DLY: 100000 ;DELAY
3787 025226 000000 DNFLAG: 0 ;DONE FLAG
3788 ;MOD U.2.3.4 ---- END MODULE -----
3789
3790 .SBTTL MOD U.2.3/4 DELAY
3791 ;-----
3792
3793 025230 000240 DELAY: NOP ;
3794 025232 023727 025330 000000 IFAU23: CMP RDYWD,#0 ;IF READY WORD
3795 025240 001430 BEQ XU23 ;EQUALS ZERO, THEN BR TO END IF 'A'
3796 025242 013704 025324 MOV RYDX,R4 ;SET READY DELAY MULT
3797 025246 013703 025326 BDAU23: MOV RYDLY,R3 ;SET READY DELAY
3798 025252 035777 025330 000052 BDBU23: BIT RDYWD,@CSRADR ;IF READY
3799 025260 001020 BNE XU23 ;EQUAL TO '1', THEN BR TO END IF 'B'
3800 025262 005303 DEC R3 ;ELSE DECREMENT DELAY
3801 025264 001372 BNE BDBU23 ;DO UNTIL R3=0
3802 025266 005304 DEC R4 ;DECREMENT DELAY MULT.
3803 025270 001366 BNE BDAU23 ;DO UNTIL R4=0
3804 025272 052737 040000 002274 BIS #40000,ERRSY ;SET TIME OUT ERR
3805 025300 017737 000026 002246 MOV @CSRADR,CSRUUT ;GET UUT C&S REG
3806 025306 062737 000002 025332 ADD #2,CSRADR ;SET CSRADR TO DB REG
3807 025314 017737 000012 002250 MOV @CSRADR,ESRUUT ;GET UUT E&S REG
3808 025322 000207 XU23: RTS PC ;RETURN TO CALLING MOD
3809 ;-----
3810 025324 000040 RYDX: 40 ;READY MULTIPLIER
3811 025326 100000 RYDLY: 100000 ;READY DELAY
3812 025330 000000 RDYWD: 0 ;READY WORD - TEST FOR DEVICE READY
3813 025332 000000 CSRADR: 0 ;C&S REG OF UNIT- WAITING FOR
3814 ;MOD U.2.3.4 ---- END MODULE -----
```

3817
3818
3819
3820
3821 025334 013737 025410 027070
3822 025342 004737 026206
3823 025346 013737 025410 030622
3824 025354 004737 030476
3825 025360 032737 020000 025410
3826 025366 001402
3827 025370 004737 025412
3828 025374 013737 025410 027626
3829 025402 004737 027230
3830 025406 000207
3831
3832 025410 000000
3833

```
.SBTTL MOD 2.4 - EVALUATE TEST RESULTS  
-----  
EVTSTR: MOV TSTEV,FUNEV ;PASS TEST FUNCTION  
          CALL EVDVST ;CALL MOD 2.4.2 - EVALUATE DRIVE STATE  
          MOV TSTEV,FNEV4 ;PASS TEST FUNCTION  
          CALL EVUTEC ;CALL MOD 2.4.4 - EVAL UNIT ERR CODE  
IFA24: BIT #20000,TSTEV ;IF DATA CK BIT  
        BEQ EIA24 ;IS SET, THEN  
        CALL EVDATA ;CALL MOD 2.4.1 - EVALUATE DATA  
EIA24: MOV TSTEV,TSTCK ;PASS DRIVE TEST  
        CALL JPDVST ;CALL MOD 2.4.3 UPDATE DRIVE STATISTICS  
        RTS PC ;  
-----  
TSTEV: 0  
:MOD 2.4 ----- END MODULE -----
```

```

3836          .SBTTL MOD 2.4.1 - EVALUATE DATA
3837          :-----
3838
3839 025412 005037 026054          EVDATA: CLR          DAERCT          :CLEAR DATA ERR COUNT
3840 025416 005037 026046          CLR          SEEKCK          :CLEAR SEEK CK
3841 025422 012737 000001 026062  MOV          #1,PTHEAD        :SET PRINT HEADER FLAG
3842 025430 013701 002252          MOV          WDCNT,R1          :SAVE WORD COUNT
3843 025434 006301          ASL          R1                :
3844 025436 162701 000001          SUB          #1,R1             :SUBTRACT 2 TO GET CHECKSUM
3845 025442 012702 034006          MOV          #DATPAT,R2        :GET ADDRESS DATA SOURCE
3846 025446 012703 034406          MOV          #DATBUF,R3        :GET ADDRESS DATA BUFFER
3847 025452 060102          ADD          R1,R2             :CAL. ADDR SOURCE CHECKSUM
3848 025454 060103          ADD          R1,R3             :CAL. ADDR BUFFER CHECKSUM
3849 025456 121213          IFA241: CMPB          (R2),(R3)  :IF CHECK SUMS
3850 025460 001407          BEQ          ELA241            :NOT= THEN
3851 025462 032737 000002 021452  IF!241: BIT          #2,ERTSAV  :IF CRC ERR
3852 025470 001003          BNE          ELA241            :NOT SET, THEN
3853 025472 052737 000004 002276  BIS          #4,ERRTY          :SET CHECKSUM ERR
3854 025500 005037 026052          ELA241: CLR          BYTNUM      :CLEAR BYTE NUMBER
3855 025504 162701 000001          SUB          #1,R1             :CAL. TOTAL BYTE COUNT-LAST TWO
3856 025510 010137 026050          MOV          R1,BYTCNT         :SAVE BYTE COUNT
3857 025514 012701 034006          BDA241: MOV          #DATPAT,R1  :SET TEMP#1=DATA SOURCE BEGIN ADR
3858 025520 012702 034406          MOV          #DATBUF,R2        :SET TEMP#2=DATA BUFFER BEGIN ADR
3859 025524 063701 026052          ADD          BYTNUM,R1         :CAL CURRENT BYTE ADDR (SOURCE)
3860 025530 063702 026052          ADD          BYTNUM,R2         :CAL CURRENT BYTE ADDR (BUFFER)
3861 025534 121112          CMPB          (R1),(R2)        :IF SOURCE BYTE & BUFFER BYTE
3862 025536 001502          BEQ          ELB241            :NOT EQUAL
3863 025540 005237 026054          INC          DAERCT            :INCREMENT DATA ERR COUNT
3864 025544 052737 000010 002276  BIS          #10,ERRTY          :SET DATA ERR-ERR TYPE
3865 025552 042737 000004 002276  BIC          #4,ERRTY          :CLR CK SUM ERR-ERR TYPE
3866 025560 023727 026052 000002  IFC241: CMP          BYTNUM,#2  :IF BYTE #0 OR #1
3867 025566 002006          BGE          IFE241            :THEN
3868 025570 005737 026052          IFD241: RST          BYTNUM      :IF BYTE #0
3869 025574 001003          BNE          IFE241            :THEN
3870 025576 052737 000001 026046  BIS          #1,SEEKCK          :SET SEEK ERR-ERR TYPE
3871 025604 023727 026054 000012  IFE241: CMP          DAERCT,#12    :IF OVER 10 DATA ERRORS
3872 025612 103404          BLO          THF241            :THEN
3873 025614 032737 000020 002204  IFF241: BIT          #20,SWREG    :IF PRINT ONLY 10 DATA ERROR FLAG
3874 025622 001047          BNE          EIF241            :IS NOT SET, THEN
3875 025624 111137 026056          THF241: MOVB          (R1),DATASB :
3876 025630 111237 026060          MOVB          (R2),DATAWS     :
3877 025634 005737 026062          IFM241: TST          PTHEAD      :IF PRINT HEADER
3878 025640 001420          BFG          EIM241            :OK, THEN
3879 025642 005037 026062          CLR          PTHEAD           :CLEAR PRINT HEADER
3880 025646          PRINTB          #DMSG1,UNIT,TRACK,SECTOR
3881 025702          PRINTB          #DMSG2,BYTNUM,<B,DATASB>,<B,DATAWS>
3882 025742 000240          EIF241: NOP
3883 025744 005237 026052          ELB241: INC          BYTNUM      :INCREMENT BYTE #
3884 025750 005337 026050          DEC          BYTCNT           :DECREMENT BYTE COUNT
3885 025754 005737 026050          TST          BYTCNT           :DO UNTIL BYTE COUNT
3886 025760 003255          BGT          BDA241            :EQUALS 0
3887 025762 005737 026046          IFJ241: TST          SEEKCK      :IF DISK SEEK ERR
3888 025766 001413          BEQ          END241            :IS SET AND
3889 025770 032737 000010 002276  IFK241: BIT          #10,ERRTY  :IF DATA ERR
3890 025776 001007          BNE          END241            :NOT SET AND
3891 026000 032737 000002 021452  IFL241: BIT          #2,ERTSAV  :IF CRC ERR

```

3892 026006 001003
3893 026010 052737 000001 002276
3894 026016 000240
3895 026020 005037 021452
3896 026024 012705 034406
3897 026030 012704 000200
3898 026034 005025
3899 026036 005304
3900 026040 005704
3901 026042 000374
3902 026044 000207
3903
3904 026046 000000
3905 026050 000000
3906 026052 000000
3907 026054 000000
3908 026056 000000
3909 026060 000000
3910 026062 000000
3911
3912 026064 047045 040445 052440
3913 026161 045 022516 031523
3914
3915

```
END241: BNE END241 ;NOT SET
        BIS #1,ERRTY ;THEN SET SEEK ERR
        NOP ;
        CLR ERTSAV ;CLEAR ERR TYP SAV
        MOV #DATBUF,R5 ;GET BEGIN OF DATA BUFFER
        MOV #128,R4 ;SET WORD LENGTH OF TABLE
BDB241: CLR (R5)+ ;CLEAR WORD IN DATA BUFFER TABLE
        DEC R4 ;DECREMENT WORD COUNT
        TST R4 ;DO UNTIL
EDB241: BNE BDB241 ;ALL TABLE WORDS ZEROED
        RTS PC ;RETURN
-----
SEEKCK: 0 ;SEEK CECK FLAG
BYTCNT: 0 ;BYTE COUNT
BYTNUM: 0 ;BYTE NUMBER
DAERCT: 0 ;DATA ERR COUNT
DATASB: 0 ;DATA SHOULD BE
DATAWS: 0 ;DATA WAS
PTHEAD: 0 ;PRINT HEADER FLAG
-----
DMSG1: .ASCIZ /%N% UNIT#%01% TRK#%D3% SEC#%D2%N% BYTE#%S2%AGOOD%S6%ABAD/
DMSG2: .ASCIZ /%N%S3%D3%S2%B8%S2%B8/
        .EVEN
;MCD 2.4.1 ----- END MODULE -----
```

```

3918 .SBTTL MOD 2.4.2 - EVALUATE DRIVE STATE
3919 -----
3920 026206 013705 002236 EVDVST: MOV UUTADR,R5
3921 026212 013737 002246 027072 MOV CSRUUT,CSREV ;GET COMMAND & STATUS LAST OP UUT
3922 026220 013737 002250 027074 MOV ESRUUT,ESREV ;GET ERROR STATUS LAST OP UUT
3923 026226 005037 033544 CLR XERUUT ;CLEAR EXTENDED ERROR CODE LOCATION
3924 026232 032737 000040 027072 IFA242: BIT #40,CSREV ;IF DONE NOT
3925 026240 001032 BNE IFB242 ;SET THEN
3926 026242 012715 040000 MOV #40000,(R5) ;ISSUE PROG INIT TO UUT
3927 026246 013737 002236 025332 MOV UUTADR,CSRADR ;SET CSR ADR
3928 026254 012737 000040 025330 MOV #DNBIT,RDYWD ;SET DONE TEST
3929 026262 004737 025230 CALL DELAY ;WAIT FOR TR
3930 026266 032715 000040 IFC242: BIT #40,(R5) ;IF DONE NOT
3931 026272 001005 BNE ELC242 ;SET THEN
3932 026274 052737 000010 002274 BIS #10,ERRSY ;SET NO DONE ON INT-SYS ERR
3933 026302 000137 027064 JMP END242 ;BR TO END MOD
3934 026306 113701 027070 ELC242: MOV#B FUNEV,R1 ;GET DRIVE FUNCTION
3935 026312 042701 177770 BIC #177770,R1 ;CLEAR ALL BUT FUNCTION
3936 026316 050137 002274 BIS R1,ERRSY ;SET NO DONE ON FUNCTION-SYS ERR
3937 026322 000137 027064 JMP END242 ;BR TO END MOD
3938 026326 004737 027076 IFB242: CALL EVDVRE ;CALL MOD 2.4.2.1 EVALUATE DRIVE RESPONSE
3939 026332 005737 002274 TST ERRSY ;IF SYS ERR
3940 026336 001463 BEQ IFG242 ;NOT=0 THEN
3941 026340 032737 000001 002234 BIT #1,UUT ;IFDRV#1 UNDER TST
3942 026346 001404 BEQ 1$ ;THEN
3943 026350 012737 000020 027066 MOV #20,EVCMD ;SET CMD TO DRV#1
3944 026356 000402 BR 2$ ;BR
3945 026360 005037 027066 1$: CLR EVCMD ;SET CMD TO DRV#0
3946 026364 052737 000013 027066 2$: BIS #13,EVCMD ;SET READ UUT ESR IN CMD
3947 026372 053737 002242 027066 BIS DEN,EVCMD ;SET DEN FOR CMD
3948 026400 013715 027066 MOV EVCMD,(R5) ;READ UUT ESR
3949 026404 013737 002236 025332 MOV UUTADR,CSRADR ;SET CSR ADR
3950 026412 012737 000040 025330 MOV #DNBIT,RDYWD ;SET DONE BIT
3951 026420 004737 025230 CALL DELAY ;CALL
3952 026424 032715 000040 IFX242: BIT #40,(R5) ;IF DONE BIT
3953 026430 001005 BNE IFD242 ;NOT SET THEN
3954 026432 052737 000200 002274 BIS #200,ERRSY ;SET NO DONF BIT (SECONDARY PROBLEM)
3955 026440 000137 027064 JMP END242 ;BR TO END
3956 026444 032715 100000 IFD242: BIT #100000,(R5) ;IF ERR BIT
3957 026450 001403 BEQ IFE242 ;SET
3958 026452 052737 100000 002276 BIS #100000,ERRTY ;ERR BIT - ERR TYPE
3959 026460 013701 002236 IFE242: MOV UUTADR,R1 ;GET UUT ADR
3960 026464 062701 000002 ADD #2,R1 ;CAL DBR ADR
3961 026470 032711 000200 BIT #200,(R1) ;IF DRV RDY BIT
3962 026474 001102 BNE IFN242 ;EQUALS 0
3963 026476 052737 000040 002274 BIS #40,ERRSY ;SET DRIVE NOT RDY-SYS ERR
3964 026504 000561 BR IFS242 ;BR TO END IF 'E'
3965 026506 032737 002021 027074 IFG242: BIT #2021,ESREV ;IF ANY ESR ERR BIT SET
3966 026514 001410 BEQ IFH242 ;THEN
3967 026516 032737 100000 027072 IFI242: BIT #100000,CSREV ;IF UUT ERR BIT
3968 026524 001010 BNE IFJ242 ;NOT=1 THEN
3969 026526 052737 040000 002276 BIS #40000,ERRTY ;SET MISSING ERR BIT
3970 026534 000450 BR IFL242 ;BR TO IF 'L'
3971 026536 032737 100000 027072 IFH242: BIT #100000,CSREV ;IF UUT CSR ERR BIT
3972 026544 001456 BEQ IFN242 ;EQUALS 1 THEN
3973 026546 013701 025410 IFJ242: MOV TSTEV,R1 ;GET TEST FUNCTION

```

3974	026552	042701	177774		BIC	#177774,R1	:CLEAR ALL BUT TWO BOTTOM BITS
3975	026556	022701	000002		CMP	#2,R1	:IF WRITE FUNCTION
3976	026562	001004			BNE	IFK242	:THEN
3977	026564	052737	004000	002276	BIS	#4000,ERRTY	:SET WRITE ERR - ERR TYPE
3978	026572	000431			BR	IF' 242	:BR TO IF 'L'
3979	026574	013701	025410		IFK242: MOV	TSTEV,R1	:GET TEST FUNCTION
3980	026600	042701	177770		BIC	#177770,R1	:CLEAR ALL BUT FUNCTION
3981	026604	022701	000003		CMP	#3,R1	:IF READ FUNCTION
3982	026610	001004			BNE	IFM242	:THEN
3983	026612	052737	002000	002276	BIS	#2000,ERRTY	:SET READ ERR-ERR TYPE
3984	026620	000416			BR	IFL242	:BR TO IF 'L'
3985	026622	013701	025410		IFM242: MOV	TSTEV,R1	:GET TEST FUNCTION
3986	026626	042701	177771		BIC	#177771,R1	:CLEAR BITS
3987	026632	032701	000006		BIT	#6,R1	:IF. FILL/EMPTY BUFFER
3988	026636	001004			BNE	ELM242	:THEN
3989	026640	052737	001000	002276	BIS	#1000,ERRTY	:SET FILL/EMPTY ERR-ERR TYP
3990	026646	000403			BR	IFL242	:BR TO IF 'L'
3991	026650	052737	000400	002276	ELM242: BIS	#400,ERRTY	:ELSE SET UNK ERR
3992	026656	032737	000001	027074	IFL242: BIT	#1,ESREV	:IF CRC ERR (ESR)
3993	026664	001406			BEQ	IFN242	:THEN
3994	026666	042737	000001	002276	BIC	#1,ERRTY	:CLEAR ANY SEEK ERR
3995	026674	052737	000002	002276	BIS	#2,ERRTY	:SET CRC ERR
3996	026702	032737	006010	027074	IFN242: BIT	#6010,ESREV	:IF ESR BIT#3,10,11 ARE
3997	026710	001404			BEQ	IFF242	:SET, THEN
3998	026712	052737	100000	002274	BIS	#100000,ERRSY	:SET UNKNOWN ERR-SYS ERR
3999	026720	000453			BR	IFS242	:BR TO IF 'S'
4000	026722	013701	025410		IFF242: MOV	TSTEV,R1	:GET TEST FUNCTION
4001	026726	032701	000002		BIT	#2,R1	:IF FUNCTION WAS
4002	026732	001446			BEQ	IFS242	:POSSIBLE READ OR WRITE
4003	026734	032701	000005		BIT	#5,R1	:BUT REALLY
4004	026740	001043			BNE	IFS242	:IS READ OR WRITE, THEN
4005	026742	005737	002244		IF0242: TST	DEL DAT	:IF DELETED DATA FLAG IS
4006	026746	001410			BEQ	IFQ242	:SET THEN
4007	026750	032737	000100	027074	IFP242: BIT	#100,ESREV	:IF UUT ESR DD BIT
4008	026756	001013			BNE	IFR242	:NOT SET THEN
4009	026760	052737	000100	002276	BIS	#100,ERRTY	:SET MISSING DP MARK-ERR TYP
4010	026766	000407			BR	IFR242	:BR TO IF 'R'
4011	026770	032737	000100	027074	IFQ242: BIT	#100,ESREV	:IF D.D. BIT IS
4012	026776	001403			BEQ	IFR242	:SET THEN
4013	027000	052737	000040	002276	BIS	#40,ERRTY	:SET UNEX DD BIT
4014	027006	032737	000020	027074	IFR242: BIT	#20,ESREV	:IF DEN. ERR. (ESR)
4015	027014	001403			BEQ	IFU242	:THEN
4016	027016	052737	020000	002274	BIS	#20000,ERRSY	:SET DEN. ERR-SYS ERR
4017	027024	005737	002242		IFU242: TST	DEN	:IF DOUBLE DEN MODE IS
4018	027030	001407			BEQ	IFS242	:SET AND THEN
4019	027032	032737	000040	027074	IFV242: BIT	#40,ESREV	:IF. UUT RESPONDS IN
4020	027040	001003			BNE	IFS242	:SINGLE DENSITY, THEN
4021	027042	052737	010000	002274	BIS	#10000,ERRSY	:SET DRIVE DENSITY ERR-SYS ERR
4022	027050	032737	100000	027072	IFS242: BIT	#100000,CSREV	:IF UUT ERR BIT
4023	027056	001402			BEQ	END242	:NOT=0 THEN
4024	027060	004737	005026		CALL	RDERCD	:CALL UUT - READ ERROR CODE
4025	027064	000207			END242: RTS	PC	
4026							
4027	027066	000000			EVCMD:	0	:CMD WORD USED IN THIS MOD
4028	027070	000000			FUNEV:	0	
4029	027072	000000			CSREV:	0	

HARDWARE TESTS MACY11 30(1046)
CZRDC.P11 29-MAR-82 14:53

29-MAR-82 15:57 PAGE 63-2
MOD 2.4.2 - EVALUATE DRIVE STATE

SEQ 0084

4030 027074 000000
4031

ESREV: 0
;MOD 2.4.2 ----- END MODULE -----

```

4034
4035
4036
4037
4038 027076 013701 025410
4039 027102 042701 177771
4040 027106 032701 000006
4041 027112 001445
4042 027114 005737 002172
4043 027120 001421
4044 027122 032737 000002 002234
4045 027130 001403
4046 027132 012701 001000
4047 027136 000401
4048 027140 005001
4049 027142 013702 002250
4050 027146 042702 176777
4051 027152 020102
4052 027154 001403
4053 027156 052737 001000 002274
4054 027164 032737 000001 002234
4055 027172 001403
4056 027174 012701 000400
4057 027200 000401
4058 027202 005001
4059 027204 013702 002250
4060 027210 042702 177377
4061 027214 020102
4062 027216 001403
4063 027220 052737 000400 002274
4064 027226 000207
4065

```

```

.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 FULL/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,ERRSY ;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,ERRSY ;SET WRONG DRIVE RESPONDING SYS ERR
6$: RTS PC
;MOD 2.4.2.1 ----- END MODULE -----

```

```
4068 .SBTTL MOD 2.4.3 - UPDATE DRIVE STATISTICS
4069 ;-----
4070
4071 027230 013737 027626 030342 UPDVST: MCV TSTCK,FUNTY ;PASS TEST FUNCTION TO UPDATE SEC CTR
4072 027236 004737 030216 CAL UPSECT ;CALL UP DATE SECTOR CONTENTS
4073 027242 032737 000002 027606 IA243: BIT #2,ETSAV ;IF ERTTY SAVE
4074 027250 001405 BEQ EA243 ;HAS CRC ERR BIT SET, THEN
4075 027252 004737 027722 CALL UDCRST ;CALL UPDATE CRC STATISTICS
4076 027256 005037 027606 CLR ETSAV ;CLEAR ERR TYPE SAVE
4077 027262 000457 BR IG243 ;BR TO IF 'G'
4078 027264 013737 002276 027606 EA243: MOV ERTTY,ETSAV ;SAVE ERR TYP --> ETSAV
4079 027272 013737 002276 027614 MOV ERTTY,STERRG ;GET ERR TYP --> STAT ERR REG
4080 027300 005037 027616 CLR STCNTR ;ZERO STAT COUNTER
4081 027304 032737 000002 027614 ID243: BIT #2,STERRG ;IF ERR IS
4082 027312 001403 BEQ BF243 ;CRC, THEN
4083 027314 042737 006002 027614 BIC #6002,STERRG ;CLEAR CRC, RD, & WRT ERR BITS OF STAT ERR REG
4084 027322 000241 BF243: CLC ;CLEAR CARRY BIT
4085 027324 006037 027614 ROR STERRG ;ROTATE RIGHT STAT ERROR REG
4086 027330 103026 IB243: BCC EB243 ;IF CARRY BIT SET, THEN
4087 027332 013701 027616 MOV STCNTR,R1 ;GET STAT COUNTER
4088 027336 006301 ASL R1 ;& DOUBLE FOR WORD ADDRESSING
4089 027340 062701 027630 ADD #ETTAB,R1 ;CAL. CLASSIFICATION WORD-ADDRESS
4090 027344 011137 027620 MOV (R1),CLASWD ;GET CLASSIFICATION WORD
4091 027350 011102 MOV (R1),R2 ;GET CLASSIFICATION WORD-TO FIND LOG OFFSET
4092 027352 000302 SWAB R2 ;GET CLASSIFICATION WORD UPPER BYTE
4093 027354 006302 ASL R2 ;--SHIFT LEFT TO GET LOG REG OFFSET (LAST 6 BITS)
4094 027356 006302 ASL R2 ;--SHIFT LEFT AGAIN
4095 027360 042702 177004 BIC #177004,R2 ;CLEAR UNWANTED BITS
4096 027364 010237 027622 MOV R2,LOGOFF ;SAVE ERROR LOG OFFSET
4097 027370 005711 IC243: TST (R1) ;IF ERR TYP CLASSIFICATION WORD
4098 027372 100403 BMI LC243 ;TYPE=SOFT, THEN
4099 027374 004737 030072 CALL UDSFST ;CALL UPDATE SOFT ERROR STATISTICS
4100 027400 000402 BR EB243 ;BR TO END 'B'
4101 027402 004737 027670 LC243: CALL UDHDST ;CALL UPDATE HARD ERROR STATISTICS
4102 027406 005237 027616 EB243: INC STCNTR ;INCREMENT STAT COUNTER
4103 027412 022737 000020 027616 UF243: CMP #16,STCNTR ;DO UNTIL ALL 16
4104 027420 101340 BHI BF243 ;BITS ARE DONE
4105 027422 013703 033544 IG243: MOV XERUUT,R3 ;GET EXTENDED ERROR CODE
4106 027426 042703 177400 BIC #177400,R3 ;CLEAR UPPER BYTE
4107 027432 005703 TST R3 ;IF EXTENDED ERROR CODE
4108 027434 001410 BEQ IH243 ;NOT=0, THEN
4109 027436 162703 000010 SUB #10,R3 ;ADJ ERROR CODE # FOR LOGGING
4110 027442 012702 007604 MOV #ECLOG,R2 ;GET LOC OF ERR CODE LOG
4111 027446 060302 ADD R3,R2 ;ADJ ERR CODE TO LOC ERR CODE LOG
4112 027450 063702 002240 ADD UUTOFF,R2 ;FIND LOC ERR REG THIS UNIT
4113 027454 005212 INC (R2) ;INCREMENT UNIT ERR REG
4114 027456 013703 002276 IH243: MOV ERTTY,R3 ;GET ERR TYPE
4115 027462 042703 171774 BIC #171774,R3 ;CLEAR ALL ERRS BUT RD, WT, CRC, SEEK
4116 027466 005703 TST R3 ;IF ONE OF THESE ERRORS
4117 027470 001412 BEQ II243 ;THEN
4118 027472 013702 002254 MOV TRACK,R2 ;GET TRACK ADR
4119 027476 006302 ASL R2 ;DOUBLE TRACK ADR FOR WORD ADDRESSING
4120 027500 006302 ASL R2 ;ADJ TRK
4121 027502 006302 ASL R2 ;FOR ADR.
4122 027504 062702 010070 ADD #TKXX,R2 ;ADD TRACK LOG LOCATION
4123 027510 063702 002240 ADD UUTOFF,R2 ;FIND LOC ERR REG THIS UNIT
```

```
4124 027514 005212          INC      (R2)          ;INCREMENT UNIT ERR REG
4125 027516 005737 027610  I1243:  TST      ERRSAV      ;IF ERR SAVE HAS
4126 027522 001023          BNE      L1243        ;NO ERROR SET, THEN
4127 027524 005237 027612  INC      ERSVCT       ;INCREMENT ERROR SAVE COUNTER
4128 027530 022737 000004 027612  IJ243:  CMP      #4,ERSVCT  ;IF ERROR SAVE COUNTER
4129 027536 101017          BHI      E1243        ;NOT=4, THEN
4130 027540 012701 002306  MOV      #SEEKRT,R1   ;SET BEGIN ADDRESS OF RETRY COUNTERS
4131 027544 012702 000011  MOV      #11,R2       ;SET # OF PENTRY COUNTERS
4132 027550 005021          BK243:  CLR      (R1)+   ;CLEAR RETRY COUNTER
4133 027552 005302          DEC      R2          ;DECREMENT RETRY COUNTER #
4134 027554 005702          UK243:  TST      R2          ;DO UNTIL
4135 027556 001374          BNE      BK243       ;ALL COUNTERS CLEARED
4136 027560 005037 027612  CLR      ERSVCT       ;CLEAR ERROR SAVE COUNTER
4137 027564 005037 002304  CLR      RETRY        ;CLEAR RETRY COUNTER
4138 027570 000402          BR      E1243        ;BR TO END 'I'
4139 027572 005037 027612  L1243:  CLR      ERSVCT       ;CLEAR ERROR SAVE COUNT
4140 027576 013737 002276 027610  E1243:  MOV      ERRTY,ERRSAV ;SAVE ERROR TYPE FOR NEXT ERROR CHECK
4141 027604 000207          END243: RTS      PC          ;RETURN
4142
4143 027606 000000          ETSAV:  0            ;ERR TYPE SAVE
4144 027610 000000          ERRSAV: 0            ;ERR TYPE SAVE REG
4145 027612 000000          ERSVCT: 0            ;ERROR SAVE COUNTER-COUNTS # OF NO ERROR PASSES
4146 027614 000000          STERRG: 0           ;STAT ERR REG
4147 027616 000000          STCNTR: 0           ;STAT COUNTER
4148 027620 000000          CLASWD: 0           ;ERROR CLASSIFICATION WORD-FROM TABLE
4149 027622 000000          LOGOFF: 0           ;ERROR LOG OFFSET FROM #CKSML
4150 027624 000000          RTOFF:  0           ;RETRY COUNTER OFFSET FROM # SEEKRT
4151 027626 000000          TSTCK:  0           ;TEST WORD-USED TO CHECK TEST DONE
4152
4153
4154
4155
4156
4157 027630 005001          ETTAB:  .WORD    005001  ;SFT /SEEK /SEEK /SK-RTMSK/ 0
4158 027632 006005          .WORD    006005  ;SFT /CRC /CRC /CRC / 1
4159 027634 100407          .WORD    100407  ;HRD /CKSML / - /HD / 2
4160 027636 012106          .WORD    012106  ;SFT /DATA /DATA /DT-RTMSK/ 3
4161 027640 154400          .WORD    154400  ;HRD / - / - / - / 4
4162 027642 113227          .WORD    113227  ;HRD /DDUNX /DD /HD / 5
4163 027644 113227          .WORD    113227  ;HRD /DDMIS /DD /HD / 6
4164 027646 154400          .WORD    154400  ;HRD / - / - / - / 7
4165 027650 154400          .WORD    154400  ;HRD /UNK / - / - / 8
4166 027652 101407          .WORD    101407  ;HRD /FIL-EMP/ - /HD / 9
4167 027654 010164          .WORD    010164  ;SFT /RD /RD /RD-RTMSK/ 10
4168 027656 011202          .WORD    011202  ;SFT /WRT /WT /WT-RTMSK/ 11
4169 027660 103407          .WORD    103407  ;HRD /INTR-ND/ - /HD / 12
4170 027662 104407          .WORD    104407  ;HRD /D-NINTR/ - /HD / 13
4171 027664 102407          .WORD    102407  ;HRD /ER-NSET/ - /HD / 14
4172 027666 154407          .WORD    154407  ;HRD /ERR BIT/ - /HD / 15
4173
4174
4175
4176
4177
4178
```

:----- ERROR TYPE CLASSIFICATION & OFFSETS TABLE -----
:TYPE/LOG-OFF/RT-OFF/CLASS /BIT#
:-----/-----/-----/-----
:-----<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)

```

4181
4182
4183
4184 027670 000240
4185 027672 032737 000007 027620
4186 027700 001007
4187 027702 013701 027622
4188 027706 062701 007354
4189 027712 063701 002240
4190 027716 005211
4191 027720 000207
4192
4193
4194
4195
4196
4197
4198
4199 027722 000240
4200 027724 032737 020000 027626
4201 027732 001425
4202 027734 032737 000010 002276
4203 027742 001007
4204 027744 012737 000020 027622
4205 027752 012737 000006 027624
4206 027760 000420
4207 027762 012737 000000 027622
4208 027770 005037 030474
4209 027774 012737 000012 027624
4210 030002 004737 030344
4211 030006 012737 000010 027622
4212 030014 012737 000006 027624
4213 030022 032737 010000 027626
4214 030030 001407
4215 030032 012737 000020 030474
4216 030040 052737 000002 030474
4217 030046 000406
4218 030050 012737 000020 030474
4219 030056 052737 000004 030474
4220 030064 004737 030344
4221 030070 000207
4222

```

```

.SBTTL MOD 2.4.3.1 - UPDATE HARD ERROR STATISTICS
-----
UDHDST: NOP
IA2431: BIT #7,CLAS4D ; IF ERROR CLASS WORD-
; CLASS=HD(7), THEN
BNE X2431 ; GET ERROR LOG OFFSET
MOV LOGOFF,R1 ; ERR LOG ADR=ERR LOG OFF + CKSML ADR
ADD #CKSML,R1 ; UUT ERR LOG ADR=UUT OFFSET + ERR LOG ADR
ADD UTOFF,R1 ; INCREMENT THE ERROR LOG
INC (R1) ; RETURN
X2431: RTS PC
:MOD 2.4.3.1 ----- END MODULE -----

```

```

.SBTTL MOD 2.4.3.2 - UPDATE CRC STATISTICS
-----
UDCRST: NOP
IA2432: BIT #BIT13,TSTCK ; IF TEST=DATA CHECK
BEQ LA2432 ; BIT SET, THEN
IB2432: BIT #BIT03,ERRTY ; IF ERR TYPE=DATA ERR
BNE LB2432 ; NOT SET, THEN
MOV #20,LOGOFF ; SET LOG OFFSET=CRC BAD LOG
MOV #6,RTOFF ; SET RETRY OFFSET=CRC ERR
BR IC2432 ; BR TO 'C'
LB2432: MOV #50,LOGOFF ; SET DATA LOG OFFSET
CLR RTMASK ; CLEAR RETRY MASK
MOV #12,RTOFF ; SET DUMMY DATA RETRY COUNTER OFFSET
CALL SFERLG ; CALL SOFT ERROR LOGGER
LA2432: MOV #10,LOGOFF ; SET LOG OFFSET=CRC ERR LOG
MOV #6,RTOFF ; SET RETRY OFFSET=CRC ERR
IC2432: BIT #BIT12,TSTCK ; IF READ AFTER WRITE (RAW)
BEQ LC2432 ; BIT SET, THEN
MOV #BIT04,RTMASK ; SET RETRY MASK=CRC
BIS #BIT1,RTMASK ; SET RETRY MASK=WRITE
BR EC2432 ; BR TO END 'C'
LC2432: MOV #BIT04,RTMASK ; SET RETRY MASK=CRC
BIS #BIT02,RTMASK ; SET RETRY MASK=READ
EC2432: CALL SFERLG ; CALL SOFT ERROR LOGGER
RETURN ; RETURN
:MOD 2.4.3.2 ----- END MODULE -----

```

4225
4226
4227
4228 030072 013702 027620
4229 030076 006202
4230 030100 006202
4231 030102 006202
4232 030104 042702 177700
4233 030110 010237 027624
4234 030114 013702 027620
4235 030120 042702 177770
4236 030124 022702 000006
4237 030130 001022
4238 030132 032737 010000 027626
4239 030140 001404
4240 030142 012737 000012 030474
4241 030150 000403
4242 030152 012737 000014 030474
4243 030160 012737 000010 027624
4244 030166 012737 000050 027622
4245 030174 000405
4246 030176 010237 030474
4247 030202 162737 000050 027622
4248 030210 004737 030344
4249 030214 000207
4250

.SBTTL MOD 2.4.3.3 - UPDATE SOFT ERROR STATISTICS

UDSFST: MOV CLASWD,R2 ;PUT CLASSIFICATION WORD IN R1
ASR R2 ; SHIFT WORD RIGHT
ASR R2 ; 3 TIMES TO GET
ASR R2 ; RETRY COUNTER OFFSET (LAST 6 BITS)
BIC #177700,R2 ;CLEAR TOP 10 BITS
MOV R2,RTOFF ;SET RETRY COUNTER OFFSET
IA2433: MOV CLASWD,R2 ;GET CLASSIFICATION WORD
BIC #177770,R2 ;CLEAR ALL BIT ERROR CLASSIFICATION
CMP #6,R2 ;IF ERROR
BNE LA2433 ;CLASS=DATA, THEN
IB2433: BIT #BIT12,TSTCK ;IF TEST HAS
BEQ LB2433 ;READ AFTER WRITE (RAW) BIT SET, THEN
MOV #12,RTMASK ;SET DATA & WRITE RETRY
BR EB2433 ;BR TO END IF 'B'
LB2433: MOV #14,RTMASK ;SET DATA & READ RETRY
EB2433: MOV #10,RTOFF ;SET DATA RT COUNTER OFFSET
MOV #50,LOGOFF ;SET DATA LOG OFFSET
BR EA2433 ;BR TO END 'A'
LA2433: MOV R2,RTMASK ;ELSE-PUT CLASS INTO RETRY MASK
SUB #50,LOGOFF ;ADJ. LOG OFFSET SO THAT 'SEK' IS LOG BEGIN
EA2433: CALL SFERLG ;CALL SOFT ERROR LOGGER
X2433: RTS PC ;RETURN
:MOD 2.4.3.3 ----- END MODULE -----

4253
4254
4255
4256
4257 030216 013701 002234
4258 030222 006301
4259 030224 006301
4260 030226 042737 177770 030342
4261 030234 022737 000003 030342
4262 030242 001002
4263 030244 005002
4264 030246 000412
4265 030250 022737 000002 030342
4266 030256 001404
4267 030260 022737 000006 030342
4268 030266 001024
4269 030270 012702 000020
4270 030274 000241
4271 030276 060102
4272 030300 005262 007314
4273 030304 100015
4274 030306 005062 007314
4275 030312 062702 000002
4276 030316 005262 007314
4277 030322 103006
4278 030324 005062 007314
4279 030330 162702 000002
4280 030334 005062 007314
4281 030340 000207
4282
4283 030342 000000
4284

```
.SBTTL MOD 2.4.3.4 - UPDATE SECTOR WRITTEN/READ COUNTERS  
:-----  
UPSECT: MOV UUT,R1 ;GET UNIT UNDER TEST  
ASL R1 ;DOUBLE FOR WORD ADDRESSING  
ASL R1 ;DOUBLE FOR 2 WORD ADDRESSING  
BIC #17,,70,FUNTY ;CLEAR ALL BUT FUNCTION  
1A2434: CMP #3,FUNTY ;IF FUNCTION TYPE  
BNE IB2434 ;IS READ, THEN  
CLR R2 ;CLEAR R2  
BR EA2434 ;BR TO END 'A'  
IB2434: CMP #2,FUNTY ;IF FUNCTION TYPE  
BEQ LB2434 ;IS NOT WRITE #1, THEN  
IC2434: CMP #6,FUNTY ;IF FUNCTION TYPE  
BNE XUPSCT ;IS WRITE #2, THEN  
LB2434: MOV #20,R2 ;SET R2 OFFSET=WRITE  
EA2434: CLC ;CLEAR CARRY BIT  
ADD R1,R2 ;SETUP OFFSET  
INC READSC(R2) ;INCREMENT SECTOR COUNTER  
BPL XUPSCT ;IF BIT#15 SET, THEN  
CLR READSC(R2) ;CLEAR SECTOR COUNTER  
ADD #2,R2 ;SETUP TO INCREMENT DOUBLE PRECISION WORD  
INC READSC(R2) ;INCREMENT DOUBLE PRECISION WORD  
BCC XUPSCT ;IF CARRY BIT SET, THEN  
CLR READSC(R2) ;CLEAR DOUBLE PRECISION CTR  
SUB #2,R2 ;  
CLR READSC(R2) ;CLEAR DOUBLE PRECISION CTR  
XUPSCT: RETURN ;RETURN  
:-----  
FUNTY: 0 ;STATISTICS FUNCTION CK  
:-----
```

4287
4288
4289
4290 030344 013701 027622
4291 030350 013702 027624
4292 030354 062702 002306
4293 030360 032737 000004 002204
4294 030366 001004
4295 030370 032737 000004 002264
4296 030376 001412
4297 030400 021227 000012
4298 030404 103007
4299 030406 005212
4300 030410 053737 030474 002304
4301 030416 005037 002300
4302 030422 000413
4303 030424 062701 007514
4304 030430 063701 002240
4305 030434 005211
4306 030436 043737 030474 002304
4307 030444 005012
4308 030446 005237 002300
4309 030452 013701 027622
4310 030456 062701 007424
4311 030462 063701 002240
4312 030466 005211
4313 030470 000240
4314 030472 000207
4315
4316 030474 000000
4317

```
.SBTTL - MOD 2.4.U.1 - SOFT ERROR LOGGER  
-----  
SFERLG: MOV LOGOFF,R1 :GET ERR LOG OFFSET  
MOV RTOFF,R2 :GET RETRY COUNTER OFFSET  
ADD #SEEKRT,R2 :CAL. RETRY COUNTER ADR  
IA24U1: BIT #BIT02,SWREG :IF (SFT SW REG) RETRY ON ERROR, LOG SOFT OR HD  
BNE IB24U1 :SET OR  
BIT #EVL,FLGDRS :DRS 'EVL' FLAG  
BEQ LB24U1 :SET, THEN  
IB24U1: CMP (R2),#12 :IF RETRY COUNTER  
BHIS LB24U1 :EQUALS < 10 ERRORS, THEN  
INC (R2) :INCREMENT RETRY COUNTER  
BIS RTMASK,RETRY :SET RT FLAGS PER RT MASK  
CLR HARDER :CLEAR HARD ERROR  
BR EB24U1 :BR TO END 'B'  
LB24U1: ADD #HSEK,R1 :HD ERR LOG ADR=HARD SEEK ADR+LOG OFFSET  
ADD UUTOFF,R1 :UUT ERR LOG ADR=UUT OFFSET+LOG ADR  
INC (R1) :INCREMENT UUT HARD ERROR LOG  
BIC RTMASK,RETRY :CLEAR RETRY FALGS USING RT MASK  
CLR (R2) :CLEAR RETRY COUNTER  
INC HARDER :SET HARD ERROR FLAG  
EB24U1: MOV LOGOFF,R1 :GET ERR LOG OFFSET  
ADD #SEK,R1 :ERR LOG ADR=SEK LOG ADR+LOG OFFSET  
ADD UUTOFF,R1 :UUT ERR LOG ADR=UUT OFFSET+LOG ADR  
INC (R1) :INCREMENT UUT ERROR LOG  
X24U1: NOP  
RTS PC :RETURN  
-----  
RTMASK: 0 :RETRY MASK  
:MOD 2.4.U1 ----- END MODULE -----
```



```

4320      .SBTTL MOD 2.4.4 - EVALUATE UNIT ERROR CODE
4321      :-----
4322
4323      030476 013701 033544      EVUTEC: MOV      XERUUT,R1      ;GET ERR CODE & SAVE
4324      030502 042701 177400      BIC      #177400,R1      ;CLEAR TOP BYTE
4325      030506 005701      IFA244: TST      R1      ;IF ERRCODE
4326      030510 001443      BEQ      END244      ;NOT=0, THEN
4327      030512 006201      ASR      R1      ;SHIFT ERR CODE FOR LOOK UP
4328      030514 006201      ASR      R1      ;AND ADDRESSING
4329      030516 062701 030624      ADD      #ECCLAS,R1      ;CAL ERR TABLE CLASSIFICATION ADH
4330      030522 011102      MOV      (R1),R2      ;GET ERR CODE CLASSIFICATION WORD
4331      030524 105702      IFB244: TSTB     R2      ;IF LOWER BYTE
4332      030526 001003      BNE      IFC244      ;EQUALS 0, THEN
4333      030530 050237 002274      BIS      R2,ERRSY      ;SET ERR ONTO ERRSY
4334      030534 000431      BR       END244      ;BR TO END IF 'B'
4335      030536 122702 000300      IFC244: CMPB     #300,R2      ;IF LOW BYTE
4336      030542 001024      BNE      ELC244      ;EQUALS 300, THEN
4337      030544 022737 000003 030622      IFD244: CMP      #3,FNEV4      ;IF FUNCTION WAS
4338      030552 001004      BNE      IFE244      ;A READ, THEN
4339      030554 052737 002000 002276      BIS      #2000,ERRTY      ;SET READ ERR
4340      030562 000416      BR       END244      ;BR TO END IF 'B'
4341      030564 022737 000002 030622      IFE244: CMP      #2,FNEV4      ;IF FUNCTION WAS
4342      030572 001004      BNE      ELE244      ;A WRITE, THEN
4343      030574 052737 004000 002276      BIS      #4000,ERRTY      ;SET WRITE ERROR
4344      030602 000406      BR       END244      ;BR TO END IF 'B'
4345      030604 052737 040000 002276      ELE244: BIS      #40000,ERRTY      ;SET UNK ERROR
4346      030612 000402      BR       END244      ;BR TO END IF 'B'
4347      030614 050237 002276      ELC244: BIS      R2,ERRTY      ;SET CLASSIFIED ERROR ONTO ERRTY
4348      030620 000207      END244: RTS      PC      ;RETURN
4349
4350      030622 000000      FNEV4: 0      ;FUNCTION FOR EVALUATION
4351
4352      030624 000000      ECCLAS: .WORD    0      ;ERR CODE # 00 ----> NOT USED (NO ERROR)
4353      030626 000001      .WORD    1      ;ERR CODE # 10 ----> SEEK
4354      030630 000001      .WORD    1      ;ERR CODE # 20 ----> SEEK
4355      030632 000000      .WORD    0      ;ERR CODE # 30 ----> NOT ASSIGNED
4356      030634 004000      .WORD    4000     ;ERR CODE # 40 ----> SYS ERR
4357      030636 000001      .WORD    1      ;ERR CODE # 50 ----> SEEK
4358      030640 002000      .WORD    2000     ;ERR CODE # 60 ----> SELF DIAG ERR
4359      030642 000300      .WORD    300      ;ERR CODE # 70 ----> READ OR WRITE ERR
4360      030644 004000      .WORD    4000     ;ERR CODE # 100 ----> SYS ERR
4361      030646 000300      .WORD    300      ;ERR CODE # 110 ----> READ OR WRITE ERR
4362      030650 000300      .WORD    300      ;ERR CODE # 120 ----> READ OR WRITE ERR
4363      030652 000300      .WORD    300      ;ERR CODE # 130 ----> READ OR WRITE ERR
4364      030654 000002      .WORD    2      ;ERR CODE # 140 ----> CRC ERR
4365      030656 000001      .WORD    1      ;ERR CODE # 150 ----> SEEK ERR
4366      030660 000300      .WORD    300      ;ERR CODE # 160 ----> READ OR WRITE ERR
4367      030662 000300      .WORD    300      ;ERR CODE # 170 ----> READ OR WRITE ERR
4368      030664 000002      .WORD    2      ;ERR CODE # 200 ----> CRC ERR
4369      030666 000000      .WORD    0      ;ERR CODE # 210 ----> NOT ASSIGNED
4370      030670 002000      .WORD    2000     ;ERR CODE # 220 ----> SELF DIAG ERR
4371      030672 004000      .WORD    4000     ;ERR CODE # 230 ----> SYS ERR
4372      030674 020000      .WORD    20000    ;ERR CODE # 240 ----> DENSITY ERR
4373      030676 020000      .WORD    20000    ;ERR CODE # 250 ----> DENSITY ERR
4374      030700 000000      .WORD    0      ;ERR CODE # 260 ----> NOT ASSIGNED
4375
:MOD 2.4.4 ----- END MODULE -----

```

```

4378 .SBTTL MOD 2.5 - OUTPUT ERROR TYPE
4379 -----
4380 030702 013737 002276 002604 OTERTP: MOV ERRTY,ERRREG ;SET ERROR TYPE FOR PRINT OUT
4381 030710 013701 002276 MOV ERRTY,R1 ;GET ERROR TYPE
4382 030714 005002 CLR R2 ;CLEAR ERROR # COUNT
4383 030716 000240 BDA25: NOP ;
4384 030720 032701 000001 IFA25: BIT #1,R1 ;IF BIT #1
4385 030724 001405 BEQ ELA25 ;EQUALS 1, THEN
4386 030726 010204 MOV R2,R4 ;SAVE ERROR # COUNT
4387 030730 006304 ASL R4 ;DOUBLE ERR # COUNT FOR ADDRESSING
4388 030732 062704 031616 ADD #ET1,R4 ;SET ADDR FOR ERR MSG PRINT
4389 030736 000407 BR THA25 ;BR TO THEN 'A'
4390 030740 000241 ELA25: CLC ;CLEAR CARRY BIT
4391 030742 006201 ASR R1 ;SHIFT ERR TYPE RIGHT
4392 030744 005202 INC R2 ;INCREMENT ERROR # COUNT
4393 030746 022702 000017 CMP #17,R2 ;DO UNTIL ERROR # COUNT
4394 030752 001361 BNE BDA25 ;EQUALS 15, THEN
4395 030754 000507 BR EIA25 ;BR TO END IF 'A'
4396 030756 005003 THA25: CLR R3 ;CLEAR R3
4397 030760 010205 MOV R2,R5 ;GET ERR#
4398 030762 062705 031656 ADD #ETCLAS,R5 ;CAL. ERR# CLASSIFICATION ADR
4399 030766 111503 MOVFB (R5),R3 ;GET ERR# CLASSIFICATION
4400 030770 032703 000001 IFB25: BIT #1,R3 ;IF SOFT ERR
4401 030774 001415 BEQ IFC25 ;CLASS, THEN
4402 030776 005737 002300 TST HARDER ;IF HARD ERR
4403 031002 001015 BNE ELB25 ;NOT SET, THEN
4404 031004 010237 002376 MOV R2,ERRNBR ;SET ERR #
4405 031010 011437 002400 MOV (R4),ERRMSG ;SET ERR MSG
4406 031014 012737 000003 002374 MOV #SOFT,ERRTYP ;SET ERRTYP=SOFT
4407 031022 004737 002354 CALL ERROR ;CALL ERROR
4408 031026 000437 BR EIC25 ;
4409 031030 032703 000002 IFC25: BIT #2,R3 ;IF HARD ERR
4410 031034 001434 BEQ EIC25 ;CLASS, THEN
4411 031036 052702 000040 ELB25: BIS #40,R2 ;SET HARD ERROR #
4412 031042 010237 002376 MOV R2,ERRNBR ;SET ERR #
4413 031046 011437 002400 MOV (R4),ERRMSG ;SET ERR MSG
4414 031052 012737 000002 002374 MOV #HARD,ERRTYP ;PRESET ERRTYP=HARD ERR
4415 031060 032737 000004 002264 IFF25: BIT #EVL,FLGDRS ;IF DRS 'EVL' FLAG
4416 031066 001413 BEQ EIF25 ;IS SET, THEN
4417 031070 005237 002302 INC HDERCT ;INCREMENT HARD ERROR CTR
4418 031074 023737 002302 002216 IFE25: CMP HDERCT,DFTL ;IF DEVICE FATAL THRESHOLD
4419 031102 101005 BHI EIF25 ;REACHED, THEN
4420 031104 012737 000001 002374 MOV #DVFT,ERRTYP ;RESET ERRTYP=DEVICE FATAL
4421 031112 005037 002302 CLR HDERCT ;CLEAR HARD ERROR CTR
4422 031116 004737 002354 EIF25: CALL ERROR ;CALL ERROR
4423 031122 005237 002300 INC HARDER ;SET HARD ERROR FLAG
4424 031126 013737 002276 002604 EIC25: MOV ERRTY,ERRREG ;SET ERR TYPE FOR PRINT OUT
4425 031134 004737 002404 CALL PRERR ;CALL U.P.ERR - PRINT ERR INFO
4426 031140 013737 002276 021452 MOV ERRTY,ERTSAV ;SAVE ERR TYP FOR DATA CK
4427 031146 005037 002276 CLR ERRTY ;CLEAR DEVICE ERR
4428 031152 004737 003034 CALL XERPRT ;CALL MOD U.PRT.B - PRINT ERR CODE
4429 031156 005737 002300 IFD25: TST HARDER ;IF NOT A
4430 031162 001002 BNE ELD25 ;HARDER, THEN
4431 031164 004737 031676 CALL PTRTY ;CALL 2.5.1 - PRINT RETRY #
4432 031170 005037 002300 ELD25: CLR HARDER ;CLEAR HARD ERROR FLAG
4433 031174 000207 EIA25: RTS PC ;RETURN

```

```

4434
4435
4436 031176 051440 042505 020113
4437 031210 041440 041522 042440
4438 031221 040 045503 051740
4439 031235 040 040504 040524
4440 031247 040 047125 051501
4441 031263 040 042504 027114
4442 031315 040 042504 027114
4443 031344 052440 040516 051523
4444 031360 052440 045516 042440
4445 031371 040 044506 046114
4446 031423 040 042522 042101
4447 031435 040 051127 052111
4448 031450 044440 052116 051105
4449 031507 040 047504 042516
4450 031546 042440 051122 051117
4451 031601 040 051105 020122
4452
4453 031616 031176
4454 031620 031210
4455 031622 031221
4456 031624 031235
4457 031626 031247
4458 031630 031263
4459 031632 031315
4460 031634 031344
4461 031636 031360
4462 031640 031371
4463 031642 031423
4464 031644 031435
4465 031646 031450
4466 031650 031507
4467 031652 031546
4468 031654 031601
4469
4470
4471
4472 031656 001
4473 031657 001
4474 031660 002
4475 031661 001
4476 031662 000
4477 031663 002
4478 031664 002
4479 031665 000
4480 031666 002
4481 031667 002
4482 031670 001
4483 031671 001
4484 031672 002
4485 031673 002
4486 031674 002
4487 031675 002
4488
4489
  
```

```

-----
ERT1: .ASCIZ / SEEK ERR/
ERT2: .ASCIZ / CRC ERR/
ERT3: .ASCIZ / CK SUM ERR/
ERT4: .ASCIZ / DATA ERR/
ERT5: .ASCIZ / UNASSG ERR/
ERT6: .ASCIZ / DEL. DATA UNEXPECTED ERR/
ERT7: .ASCIZ / DEL. DATA MISSING ERR/
ERT8: .ASCIZ / UNASSG ERR/
ERT9: .ASCIZ / UNK ERR/
ERT10: .ASCIZ / FILL OR EMPTY BUFFER ERR/
ERT11: .ASCIZ / READ ERR/
ERT12: .ASCIZ / WRITE ERR/
ERT13: .ASCIZ / INTERRUPT BUT NO DONE BIT ERR/
ERT14: .ASCIZ / DONE BIT BUT NO INTERRUPT ERR/
ERT15: .ASCIZ / ERROR, BUT NO ERR BIT SET/
ERT16: .ASCIZ / ERR BIT SET/

.EVEN
ET1: .WORD ERT1
      .WORD ERT2
      .WORD ERT3
      .WORD ERT4
      .WORD ERT5
      .WORD ERT6
      .WORD ERT7
      .WORD ERT8
      .WORD ERT9
      .WORD ERT10
      .WORD ERT11
      .WORD ERT12
      .WORD ERT13
      .WORD ERT14
      .WORD ERT15
      .WORD ERT16
  
```

```

-----
:ERROR - TYPE - ERR#
-----
ETCLAS: .BYTE 1 :SEEK - SOFT - 0 -32
        .BYTE 1 :CRC - SOFT - 1 -33
        .BYTE 2 :CKSUM - HARD - -34
        .BYTE 1 :DATA - SOFT - 3 -35
        .BYTE 0 :UNASSIGNED -
        .BYTE 2 :DEL. DATA UNEX - HARD - -37
        .BYTE 2 :DEL. DATA MISSING - HARD - -33
        .BYTE 0 :UNASSIGNED -
        .BYTE 2 :UNK ERR - HARD - -40
        .BYTE 2 :FILL/EMPTY BUFFER - HARD - -41
        .BYTE 1 :READ - SOFT - 10-42
        .BYTE 1 :WRITE - SOFT - 11-43
        .BYTE 2 :INTER-BUT NO DONE - HARD - -44
        .BYTE 2 :DONE-BUT NO INTER - HARD - -45
        .BYTE 2 :ERR-BUT NO ERR BIT - HARD - -46
        .BYTE 2 :ERR BIT SET - HARD - -47
        .EVEN
,MOD 2.5 ----- END MODULE -----
  
```

4492
4493
4494
4495
4496 031676 000240
4497 031700 005737 002304
4498 031704 001500
4499 031706 032737 000001 002304
4500 031714 001405
4501 031716 013702 002306
4502 031722 012701 032110
4503 031726 000465
4504 031730 032737 000002 002304
4505 031736 001427
4506 031740 032737 000030 002304
4507 031746 001416
4508 031750 032737 000010 002304
4509 031756 001405
4510 031760 013702 002316
4511 031764 012701 032134
4512 031770 000444
4513 031772 013702 002314
4514 031776 012701 032270
4515 032002 000437
4516 032004 013702 002326
4517 032010 012701 032166
4518 032014 000432
4519 032016 032737 000004 002304
4520 032024 001430
4521 032026 032737 000030 002304
4522 032034 001416
4523 032036 032737 000010 002304
4524 032044 001405
4525 032046 013702 002316
4526 032052 012701 032213
4527 032056 000411
4528 032060 013702 002314
4529 032064 012701 032321
4530 032070 000404
4531 032072 013702 002324
4532 032076 012701 032244
4533 032102 004737 004536
4534 032106 000207
4535
4536 032110 040445 051440 042505
4537 032134 040445 042040 052101
4538 032166 040445 053440 044522
4539 032213 045 020101 040504
4540 032244 040445 051040 040505
4541 032270 040445 041440 041522
4542 032321 045 020101 051103
4543 032352
4544

.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,R2 ;SET SEEK RT COUNT
MOV #MSKRT,R1 ;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,R2 ;SET DATA RT COUNT
MOV #MDWTRT,R1 ;SET DATA WRT MSG
BR EIB251 ;BR TO END IF 'B'
ELG251: MOV CRCRT,R2 ;SET CRC RETRY COUNT
MOV #MCWTRT,R1 ;SET CRC WRT MSG
BR EIB251 ;BR TO END IF 'B'
ELD251: MOV WRTRT,R2 ;SET WRT RT COUNT
MOV #MWTRT,R1 ;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,R2 ;SET DATA RT COUNT
MOV #MDRDRT,R1 ;SET DATA READ RT MSG
BR EIB251 ;BR TO END IF 'B'
ELH251: MOV CRCRT,R2 ;SET CRC RETRY COUNT
MOV #MCRDRT,R1 ;SET CRC READ MSG
BR EIB251 ;BR TO END IF 'B'
ELF251: MOV READRT,R2 ;SET READ RT COUNT
MOV #MRDRT,R1 ;SET READ RT MSG
EIB251: CALL PRTB1S ;PRINT RETRY # & TYPE
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/
MDRDRT: .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 -----

```
4547          .SBTTL MOD 2.6 - SET DRIVES DONE
4548          ;-----
4549
4550 032352 000240          STDVDN: NOP          ;
4551 032354 005737 021442 IFA26: TST          DVDNCK          ; IF DRV DONE CK
4552 032360 001430          BEQ          END26          ; IS SET, THEN
4553 032362 000240          NOP          ;
4554 032364 005037 021442          CLR          DVDNCK          ; CLEAR DRV DONE CK
4555 032370 032737 000001 002234 IFB26: BIT          #1,UUT          ; IF DRV#1 DONE
4556 032376 001404          BEQ          ELB26          ; THEN
4557 032400 052737 000002 021432          BIS          #2,BTHDRV          ; SET DRV#1 DONE FLAG
4558 032406 000403          BR          EIB26          ; BR TO END
4559 032410 052737 000001 021432 ELB26: BIS          #1,BTHDRV          ; SET DRV#0 DONE FLAG
4560 032416 005001          EIB26: CLR          R1          ; CLEAR TEMP DRV DONE REG
4561 032420 013703 002234          MOV          UUT,R3          ; GET UNIT UNDER TEST
4562 032424 000261          SEC          ; SET CARRY BIT
4563 032426 006101          BDA26: ROL          R1          ; MOVE DRV BIT
4564 032430 005303          DEC          R3          ; DECREMENT UNIT UNDER TEST
4565 032432 005703          TST          R3          ; DO UNTIL UNIT UNDER TST
4566 032434 002374          DUA26: BGE          BDA26          ; EQUALS -1
4567 032436 050137 021444          BIS          R1,DRVDN          ; THEN SET THIS DRV DONE
4568 032442 000207          END26: RTS          PC          ; RETURN
4569          ;MOD 2.6 ---- END MODULE -----
```

4572
4573
4574 032444 000240
4575 032446 023737 002232 002230
4576 032454 001003
4577 032456 012737 000001 014020
4578 032464 000207
4579

```
.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 -----
```

```
4582 .SBTTL MOD 4.0 - OUTPUT SYSTEM ERROR
4583 ;-----;
4584 032466 013701 002274 OTSYER: MOV ERRSY,R1 ;GET SYSTEM ERR
4585 032472 000241 CLC ;CLEAR CARRY BIT
4586 032474 006201 ASR R1 ;SHIFT
4587 032476 000241 CLC ;
4588 032500 006201 ASR R1 ; FUNCTION
4589 032502 006201 ASR R1 ; OUT
4590 032504 005002 CLR R2 ;CLEAR ERR # COUNT
4591 032506 000240 BDA40: NOP ;
4592 032510 032701 000001 IFA40: BIT #1,R1 ;IF BIT #1
4593 032514 001405 BEQ ELA40 ;EQUALS 1, THEN
4594 032516 010204 MOV R2,R4 ;SAVE ERROR # COUNT
4595 032520 006304 ASL R4 ;DOUBLE ERR # COUNT FOR ADDRESSING
4596 032522 062704 033416 ADD #SE1,R4 ;SET ADDR FOR ERR MSG PRINT
4597 032526 000406 BR THA40 ;BR TO THEN 'A'
4598 032530 006201 ELA40: ASR R1 ;SHIFT ERR TYPE RIGHT
4599 032532 005202 INC R2 ;INCREMENT ERROR # COUNT
4600 032534 022702 000017 CMP #17,R2 ;DO UNTIL ERR # COUNT
4601 032540 001362 BNE BDA40 ;EQUALS 15, THEN
4602 032542 000452 BR EIA40 ;BR TO END IF 'A'
4603 032544 010205 THA40: MOV R2,R5 ;GET ERR#
4604 032546 062705 033450 ADD #ESCLAS,R5 ;CAL. ERR# CLASSIFICATION ADR
4605 032552 111503 MOVB (R5),R3 ;GET ERR# CLASSIFICATION
4606 032554 032703 000002 IFB40: BIT #2,R3 ;IF DEVICE FATAL
4607 032560 001415 BEQ IFC40 ;ERROR, THEN
4608 032562 010205 MOV R2,R5 ;GET ERR#
4609 032564 052705 000100 BIS #100,R5 ;SET ERR CLASS=SYS
4610 032570 010537 002376 MOV R5,ERRNBR ;SET ERR#
4611 032574 011437 002400 MOV (R4),ERRMSG ;SET ERR MSG
4612 032600 012737 000001 002374 MOV #DVFT,ERRTYP ;SET DEVICE FATAL ERROR
4613 032606 004737 002354 CALL ERROR ;CALL ERROR
4614 032612 000417 BR EIC40 ;BR TO END IF 'C'
4615 032614 032703 000004 IFC40: BIT #4,R3 ;IF SYSTEM FATAL
4616 032620 001414 BEQ EIC40 ;ERROR, THEN
4617 032622 010205 MOV R2,R5 ;GET ERR#
4618 032624 052705 000200 BIS #200,R5 ;SET ERR CLASS=SYS
4619 032630 010537 002376 MOV R5,ERRNBR ;SET ERR#
4620 032634 011437 002400 MOV (R4),ERRMSG ;SET ERR MSG
4621 032640 012737 000000 002374 MOV #SYFT,ERRTYP ;SET ERR TYP=SYS FATAL
4622 032646 004737 002354 CALL ERROR ;CALL ERROR
4623 032652 013737 002274 002604 EIC40: MOV ERRSY,ERRREG ;SET SYS ERR FOR PRINT OUT
4624 032660 004737 002404 CALL PRTERR ;CALL U.P.ERR - PRINT ERR INFO
4625 032664 004737 003034 CALL XERPRT ;CALL MCD U.PRT.B - PRINT ERROR CODE
4626 032670 000240 EIA40: NOP ;
4627 032672 005037 002274 CLR ERRSY ;CLEAR SYS ERRORS
4628 032676 000207 END40: RTS PC
4629 ;-----;
```

```

4632
4633
4634 032700 047040 020117 047504
4635 032733 040 047516 042040
4636 032764 047040 020117 051104
4637 033010 047040 020117 044523
4638 033033 040 047516 042040
4639 033072 053440 047522 043516
4640 033122 053440 047522 043516
4641 033151 040 047125 051525
4642 033161 040 047125 051525
4643 033171 040 044504 045523
4644 033225 040 042504 051516
4645 033242 052040 046511 020105
4646 033302 052440 041516 040514
4647 033335 045 022516 043101
4648 033363 045 022516 051501
4649 033416
4650 033416 032700
4651 033420 032733
4652 033422 032764
4653 033424 033010
4654 033426 033033
4655 033430 033072
4656 033432 033122
4657 033434 033151
4658 033436 033101
4659 033440 033171
4660 033442 033225
4661 033444 033242
4662 033446 033302
4663
4664
4665
4666 033450 004
4667 033451 002
4668 033452 002
4669 033453 002
4670 033454 004
4671 033455 004
4672 033456 000
4673 033457 000
4674 033460 002
4675 033461 002
4676 033462 004
4677 033463 004
4678 033464 004
4679 033466
4680

```

```

-----
SYSE4: .ASCIZ / NO DONE BIT ON INITIALIZE/
SYSE5: .ASCIZ / NO DONE BIT ON FUNCTION/
SYSE6: .ASCIZ / NO DRIVE READY BIT/
SYSE7: .ASCIZ / NO SIDE READY BIT/
SYSE8: .ASCIZ / NO DONE BIT AFTER READ STATUS/
SYSE9: .ASCIZ / WRONG DRIVE RESPONDING/
SYSE10: .ASCIZ / WRONG SIDE RESPONDING/
SYSE11: .ASCIZ / UNUSED/
SYSE12: .ASCIZ / UNUSED/
SYSE13: .ASCIZ / DISKETTE WRONG DENSITY ERR/
SYSE14: .ASCIZ / DENSITY ERR/
SYSE15: .ASCIZ / TIME OUT ON 'TR' OR 'DONE' BIT/
SYSE16: .ASCIZ / UNCLASSIFIED SYSTEM ERROR/
FUNCT: .ASCIZ /%N%AFUNCTION CODE:%O3/
ERRORS: .ASCIZ /%N%ASYSTEM ERROR REG=%B%N/
-----
SE1: .EVEN
      .WORD SYSE4
      .WORD SYSE5
      .WORD SYSE6
      .WORD SYSE7
      .WORD SYSE8
      .WORD SYSE9
      .WORD SYSE10
      .WORD SYSE11
      .WORD SYSE12
      .WORD SYSE13
      .WORD SYSE14
      .WORD SYSE15
      .WORD SYSE16

```

```

-----
:ERROR - CLASS -ERR#
-----
ESCLAS: .BYTE 4 :NO DONE ON INIT - SYS FATAL - 128
        .BYTE 2 :NO DONE ON FUNCTION - DEV FATAL - 65
        .BYTE 2 :NO DRIVE RDY - DEV FATAL - 66
        .BYTE 2 :NO SIDE RDY - DEV FATAL - 67
        .BYTE 4 :NO DONE AFTER RD STA - DEV FATAL - 68
        .BYTE 4 :WRG DRV RESPOND - SYS FATAL - 133
        .BYTE 0 :WRG SIDE RESPOND - SYS FATAL - 134
        .BYTE 0 :UNUSED - 0
        .BYTE 2 :UNUSED - 0
        .BYTE 2 :DISKETT WRG DEN - DEV FATAL - 73
        .BYTE 4 :DENSITY ERR - DEV FATAL - 74
        .BYTE 4 :T.O. ON 'TR' OR 'DONE' - SYS FATAL - 139
        .BYTE 4 :SYS ERR - SYS FATAL - 140
        .EVEN
:MOD 4.0 ----- END MODULE -----

```


4683
4684
4685 033466 013737 002220 033542
4686 033474 004737 033516
4687 033500 000002
4688
4689
4690
4691
4692 033502 013737 002222 033542
4693 033510 004737 033516
4694 033514 000002
4695
4696
4697
4698
4699 033516 012737 000001 025226
4700 033524 013701 033542
4701 033530 012137 002246
4702 033534 011137 002250
4703 033540 000207
4704
4705 033542 000000
4706
4707
4708
4709
4710 033544 000
4711 033545 000
4712 033546 000
4713 033547 000
4714 033550 000
4715 033551 000
4716 033552 000
4717 033553 000
4718
4719
4720
4721
4722 033554 000232
4725
4726
4727
4728
4729 034006 000400
4732 034406 000400
4735
4736 035006

```
.SBTTL - MOD INTR.1 - INTERRUPT HANDLER #0
-----
INTH0: MOV UOADR,INCSAD ;SET UNIT #0 ADDRESS
      CALL SVUTRG ;CALL MOD U.INTR.U - SAVE UNIT REG
      RTI ;
;MOD U.INTR.1 ----- END MODULE -----

.SBTTL - MOD INTR.2 - INTERRUPT HANDLER #1
-----
INTH1: MOV U1ADR,INCSAD ;SET UNIT #1 ADDRESS
      CALL SVUTRG ;CALL MOD U.INTR.U - SAVE UNIT REG
      RTI ;
;MOD U.INTR.2 ----- END MODULE -----

.SBTTL MOD U.INTR.U - SAVE UNIT REG
-----
SVUTRG: MOV #1,DNFLAG ;SET DONE FLAG
        MOV INCSAD,R1 ;SAVE UUT ADDRESS
        MOV (R1)+,CSRUUT ;SAVE UUT CSR
        MOV (R1),ESRUUT ;SAVE UUT ESR
        RTS PC ;RETURN
-----
INCSAD: 0 ;INTERRUPTING UNIT CSR ADDRESS
;MOD U.I.U ----- END MODULE -----

.SBTTL - READ ERROR CODE BUFFER
-----
XERUUT: .BYTE 0 ;ERROR CODE UUT
WC: .BYTE 0 ;WORD COUNT UUT
CTK0: .BYTE 0 ;CUR TRK DRV#0
CTK1: .BYTE 0 ;CUR TRK DRV#1
TRK: .BYTE 0 ;TARGET TRK
TSEC: .BYTE 0 ;TARGET SEC
SFTSTS: .BYTE 0 ;MICRO CODE SOFT STATUS
BTRK: .BYTE 0 ;BAD TRK ADR
-----

.SBTTL - TRACK TABLE
-----
TRKTBL:
-----

.SBTTL - DATA BUFFERS
-----
DATPAT:
DATBUF:
-----
ENDTST
```

4739
4750
4751
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797 035010
4798
4799 035012
4800 035022
4801 035032
4802 035044
4803
4804 035056
4805
4811 035060
4812
4813 035060 054122 041040 051525
4814 035073 126 041505 047524
4815 035106 051104 053111 020105
4816 035121 105 050130 053440
4817
4818

.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: .ASLIZ /RX BUS ADR/
MSG2: .ASCIZ /VECTOR ADR/
MSG3: .ASCIZ /DRIVE # /
MSG4: .ASCIZ /EXP WRD-CR/

.EVEN

4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838 035134
4839
4840 035136
4841 035144
4842 035146
4843 035154
4844 035166
4845 035200
4846 035212
4847 035224
4848 035232
4849 035240
4850 035246
4851 035250
4852 035256
4853 035264
4854 035272
4855 035300
4856 035306
4857 035310
4858 035322
4859 035334
4860 035342
4861 035344
4862 035356
4863 035370
4864 035376
4865
4872
4873
4874
4875 035400

.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
GPRMD MSG9,24,D,177777,1,,10000,,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,D,177777,0,,76,,YES
GPRMD MSG24,16,D,177777,0,,76,,YES
5\$: GPRML MSG25,2,400,YES
XFERF 6\$
GPRMD MSG26,20,D,177777,1,,26,,YES
GPRMD MSG27,22,D,177777,1,,26,,YES
6\$: GPRML MSG5,0,177777,YES
EXIT SFT

.EVEN

ENDSFT

```

4878
4879      000015
4880      000012
4881 035400 054122 054130 042440
4882 035432 042510 050114 052040
4883 035453      105 042530 041522
4884 035475      040 020040 020060
4885 035547      040 020040 020061
4886 035572 020040 031040 036440
4887 035615      040 020040 020063
4888 035653      040 020040 020064
4889 035710 020040 032440 036440
4890 035746 020040 033040 036440
4891 036030 040504 040524 050040
4892 036056 020040 030040 036440
4893 036075      040 020040 020061
4894 036113      040 020040 020062
4895 036130 020040 031440 036440
4896 036156 020040 032040 036440
4897 036203      040 020040 020065
4898 036217      040 020040 020066
4899 036233      124 040522 045503
4900 036263      040 020040 020060
4901 036302 020040 030440 036440
4902 036331      040 020040 020062
4903 036360 020040 031440 036440
4904 036426 020040 032040 036440
4905 036471      040 020040 020065
4906 036550 020040 033040 036440
4907 036621      040 020040 020040
4908 036703      055 042076 053105
4909 037016 020040 043111 042040
4910 037126 020040 044124 020105
4911 037235      124 050131 020105
4912 037263      105 042530 041522
4913 037312 040504 040524 050040
4914 037341      124 040522 045503
4915 037370 042504 044526 042503
4916 037425      122 047125 052040
4917 037463      122 047125 052040
4918 037521      101 054516 050040
4919 037557      040 020040 042522
4920 037627      040 020040 042522
4921 037677      040 020040 051120
4922 037747      040 020040 046103
4923 040017      115 042117 043111
4924 040055      040 020040 052517
4925 040105      040 020040 047111
4926 040135      115 042117 043111
4927 040173      040 020040 044515
4928 040220 020040 046440 054101
4929
4930      040246

```

```

:-----:
CR==15      ;CARRIAGE RETURN
LF==12      ;LINE FEED
MSG5: .ASCIZ /RXXX EXPANSION TYPE <CR> /
MSG6: .ASCIZ /HELP TEST SETUP /
MSG7: .ASCIZ /EXERCISE OPTIONS/<CR><LF>
      .ASCIZ / 0 = WRITE-READ-DATA CK & READ-DATA CK/<CR><LF>
      .ASCIZ / 1 = WRITE ONLY/<CR><LF>
      .ASCIZ / 2 = WRITE-READ/<CR><LF>
      .ASCIZ / 3 = WRITE-READ-DATA CHECK/<CR><LF>
      .ASCIZ / 4 = READ-DATA CHECK ONLY/<CR><LF>
      .ASCIZ / 5 = READ ONLY (CRC CHECK)/<CR><LF>
      .ASCIZ / 6 = WRITE-READ-DATA CHECK ON ALTERNATE DRIVES/<CR><LF>
      .ASCIZ /DATA PATTERN OPTIONS/<CR><LF>
      .ASCIZ / 0 = RANDOM/<CR><LF>
      .ASCIZ / 1 = ZEROS/<CR><LF>
      .ASCIZ / 2 = ONES/<CR><LF>
      .ASCIZ / 3 = FLOATING ZERO/<CR><LF>
      .ASCIZ / 4 = FLOATING ONE/<CR><LF>
      .ASCIZ / 5 = 125/<CR><LF>
      .ASCIZ / 6 = 333/<CR><LF>
      .ASCIZ /TRACK SEQUENCE OPTIONS/<CR><LF>
      .ASCIZ / 0 = RANDOM/<CR><LF>
      .ASCIZ / 1 = INCREMENT O.D./<CR><LF>
      .ASCIZ / 2 = DECREMENT I.D./<CR><LF>
      .ASCIZ / 3 = INCREMENT O.D.-DECREMENT I.D./<CR><LF>
      .ASCIZ / 4 = BOUNCE BETWEEN I.D. & O.D./<CR><LF>
      .ASCIZ / 5 = BOUNCE BETWEEN INCR. O.D. & DECR. I.D./<CR><LF>
      .ASCIZ / 6 = BOUNCE BETWEEN O.D. & DECR. I.D./<CR><LF>
      .ASCIZ / (O.D. = OUTSIDE DIA. & I.D. = INSIDE DIA.)/<CR><LF>
      .ASCIZ /->DEVICE FATAL THRESHOLD LVL=NO. OF HARD ERRS THAT CAUSE DEVICE
      .ASCIZ / IF DRS 'EVL' FLAG IS SET, BUT HARD ERR WILL STILL LOG AS A HA
      .ASCIZ / THE 'EVL' FLAG WILL CAUSE 10 RETRIED SOFT ERRS TO BECOME A HA
      .ASCIZ /TYPE "CR" TO CONTINUE/
MSG8: .ASCIZ /EXERCISE # (0-6)/
MSG11: .ASCIZ /DATA PATTERN # (0-6)/
MSG14: .ASCIZ /TRACK SEQUENCE # (0-6)/
MSG9: .ASCIZ /DEVICE FATAL THRESHOLD LEVEL/
MSG15: .ASCIZ /RUN TEST IN DOUBLE DENSITY /
MSG16: .ASCIZ /RUN TEST IN DELETED DATA MODE/
MSG17: .ASCIZ /ANY PROGRAM CONTROL FLAGS /
MSG18: .ASCIZ / RETRY ON ERROR, LOG SOFT & HARD ERRS/
MSG19: .ASCIZ / RECALIBRATE ON SEEK ERRORS /
MSG20: .ASCIZ / PRINT ONLY 10 DATA ERRORS & CONTINUE/
MSG21: .ASCIZ / CLEAR STATISTICAL TABLES NEXT PASS /
MSG22: .ASCIZ /MODIFY TRACK ADDRESS LIMITS /
MSG23: .ASCIZ / OUTER DIAMETER ADR #/
MSG24: .ASCIZ / INNER DIAMETER ADR #/
MSG25: .ASCIZ /MODIFY SECTOR ADDRESS LIMITS /
MSG26: .ASCIZ / MIN. SECTOR ADR #/
MSG27: .ASCIZ / MAX. SECTOR ADR #/
:-----:
.EVEN

```

4933
4934
4935 040246 000000
4936 040450
4937
4938
4945
4946
4947 040450
(3) 040454
4948 040454
4949
4950 040454
4951 040454
4952 040460 177170
4953 040462 000264
4954 040464 000000
4955 040466 000000
4956 040470
4957 040470
4958 040474 177170
4959 040476 000264
4960 040500 000001
4961 040502 000000
4962 040504
4963 040504
4964 000001

.SBTTL - PATCH AREA
:-----
PATCH: 0 ;PATCH AREA
.=.+200
:-----

L\$LAST:: LASTAD
ENDMOD

BGNSETUP 2
BGNPTAB
177170
264
0
0
ENDPTAB
BGNPTAB
177170
264
1
0
ENDPTAB
ENDSETUP

.END

DNFLAG	025226	3763*	3767	3787#	4699*		
DRIVEN	015270	2550*	2553*	2596#			
DRVDN	021444	3043*	3067*	307P	3089*	3129#	4567*
DRVFN	023332	3397	3431	343	3460#	3664*	
DRVTST	023324	3099*	3101	3403	3423	3429	3457#
DUA121	015246	2585#					
DUA26	032434	4566#					
DUB20	021370	3110	3112#				
DUC20	021000	3052#					
DUMSG1	013442	2228	2235#				
DUMSG2	013503	2220	2236#				
DVDNCK	021442	3042*	3128#	3168*	3174*	4551	4554*
DVFT =	000001 G	1059#	4420	4612			
DVTST	024404	3395*	3428*	3429*	3641	3667#	
DX	025222	3765	3785#				
EA1211	016004	2675	2680#				
EA243	027264	4074	4078#				
EA2433	030210	4245	4248#				
EA2434	030274	4264	4270#				
EB24U1	030452	4302	4309#				
EB243	027406	4086	4100	4102#			
EB2433	030160	4241	4243#				
ECCLAS	030624	4329	4352#				
ECLOG	007604	1635	1854#	4110			
ECTAB	003116	1286	1296#				
EC1	003170	1296	1321#				
EC10	003555	1303	1328#				
EC11	003603	1304	1329#				
EC12	003660	1305	1330#				
EC13	003714	1306	1331#				
EC14	003773	1307	1332#				
EC15	004021	1308	1333#				
EC16	004107	1309	1334#				
EC17	004153	1310	1335#				
EC2	003236	1297	1322#				
EC20	004207	1311	1336#				
EC21	004254	1312	1337#				
EC22	004311	1313	1338#				
EC23	004360	1314	1339#				
EC24	004413	1315	1340#				
EC2432	030064	4217	4220#				
EC25	004442	1316	1341#				
EC3	003304	1298	1323#				
EC4	003332	1299	1324#				
EC5	003400	1300	1325#				
EC6	003451	1301	1326#				
EC7	003477	1302	1327#				
EDB241	026042	3901#					
EDC20	021404	3054	3116#				
ED00	013746	2362	2364	2366#			
ED1211	016000	2677	2679#				
ED2341	025102	3754	3756#				
EF.CON=	000036 G	982#	1020#				
EF.NEW=	000035 G	982#	1021#				
EF.PWR=	000034 G	982#	1022#	1896			
EF.RES=	000037 G	982#	1019#	1909			

EIK234	024560	3695	3697#	
EIM241	025702	3878	3881#	
EI243	027576	4129	4138	4140#
ELA10	014054	2395	2397#	
ELA11	014120	2413	2416#	
ELA12	014270	2445	2449#	
ELA20	020774	3047	3051#	
ELA231	023462	3484	3497#	
ELA241	025500	3850	3852	3854#
ELA25	030740	4385	4390#	
ELA40	032530	4593	4598#	
ELB11	014162	2420	2423#	
ELB12	014342	2454	2458#	
ELB121	015036	2548	2552#	
ELB20	021276	3060	3098#	
ELB22	022352	3308	3311#	
ELB231	023622	3502	3527#	
ELB232	024206	3602	3614	3616#
ELB241	025744	3862	3883#	
ELB25	031036	4403	4411#	
ELB26	032410	4556	4559#	
ELC11	014206	2425	2428#	
ELC22	022400	3313	3316#	
ELC231	023620	3508	3526#	
ELC233	024340	3653	3656#	
ELC242	026306	3931	3934#	
ELC244	030614	4336	4347#	
ELD11	012300	1982	2038#	
ELD22	022422	3318	3321#	
ELD231	023610	3513	3524#	
ELD233	024362	3658	3661#	
ELD234	024714	3704	3716#	
ELD25	031170	4430	4432#	
ELD251	032004	4507	4516#	
ELE12	014556	2476	2491#	
ELE22	022450	3323	3326#	
ELE23	023130	3411	3413	3419#
ELE234	024764	3722	3725#	
ELE244	030604	4342	4345#	
ELF12	014546	2482	2489#	
ELF20	021134	3072	3074#	
ELF231	023602	3515	3521#	
ELF232	024100	3594	3597#	
ELF251	032072	4522	4531#	
ELG11	012022	1991	1995#	
ELG12	014622	2495	2499#	
ELG21	021736	3185	3189#	
ELG251	031772	4509	4513#	
ELH11	012066	2000	2004#	
ELH12	014412	2465	2468#	
ELH20	021224	3085	3088#	
ELH231	023452	3489	3494#	
ELH234	024622	3687	3705#	
ELH251	032060	4524	4528#	
ELI11	012250	2009	2035#	
ELJ21	022064	3204	3209#	

FSCLEA= 000007	785#	2057	2070										
FSDU = 000016	785#	2210	2230										
FSEND = 000041	785#	811	913	975	1348	1368	1552	1605	1649	1933	2070	2130	2230
	2270	2333	2335	2350	2356	2367	2379	2578	2680	3114	4736	4804	4811
	4864	4875	4948	4950	4951	4956	4957	4962	4963				
F\$HARD= 000004	785#	4797	4804	4811	4841	4850	4856	4860	4864				
F\$HW = 000013	785#	865	878										
F\$INIT= 000006	785#	1889	1933										
F\$JMP = 000050	785#	2379	4804	4864									
F\$MOD = 000000	785#	811	913	975	1552	1605	4948						
F\$MSG = 000011	785#	1347	1348	1366	1368								
F\$PROT= 000021	785#	1875	1879										
F\$PWR = 000017	785#												
F\$RPT = 000012	785#	1613	1649										
F\$SEG = 000003	785#	2542	2578	2612	2680	3055	3114						
F\$SOFT= 000005	785#	4838	4841	4850	4856	4860	4864	4875					
F\$SRV = 000010	785#												
F\$SUB = 000002	785#	2335	2350	2356	2367								
F\$SW = 000014	785#	890	911										
F\$TEST= 000001	785#	2333	4736										
GETSEC 023334	3421*	3467#											
GETTRK 023742	3415*	3568#											
GETTST 021454	3057*	3139#											
GPSUNO 014626	2440*	2505#											
GPSUN1 014702	2474*	2516*	2522#										
GTDRV 022320	3096*	3304#											
GTDFN 024254	3396*	3430*	3640#										
GTEX 017302	2397	2746#											
GTEXCD 014076	2393	2411#											
GTSYEX 014022	2349*	2390#											
GTSYS 014216	2396	2439#											
GTTK 020062	2894	2897#											
G\$CNTD= 000200	785#												
G\$DELM= 000372	785#												
G\$DISP= 000003	785#												
G\$EXCP= 000400	785#												
G\$HILI= 000002	785#												
G\$LOLI= 000001	785#												
G\$NO = 000000	785#												
G\$OFFS= 000400	785#	2630	4799	4800	4801	4802	4840	4842	4843	4844	4845	4846	4847
	4848	4849	4851	4852	4853	4854	4855	4857	4858	4859	4861	4862	4863
G\$OF SI= 000376	785#	2630	4799	4800	4801	4802	4840	4842	4843	4844	4845	4846	4847
	4848	4849	4851	4852	4853	4854	4855	4857	4858	4859	4861	4862	4863
G\$PRMA= 000001	785#	4799	4800										
G\$PRMD= 000002	785#	4801	4802	4843	4844	4845	4846	4857	4858	4861	4862		
G\$PRML= 000000	785#	2630	4840	4842	4847	4848	4849	4851	4852	4853	4854	4855	4859
	4863												
G\$RADA= 000140	785#												
G\$RADB= 000000	785#												
G\$RADD= 000040	785#	4846	4857	4858	4861	4862							
G\$RADL= 000120	785#	2630	4840	4842	4847	4848	4849	4851	4852	4853	4854	4855	4859
	4863												
G\$RADO= 000020	785#	4799	4800	4801	4802	4843	4844	4845					
G\$XFER= 000004	785#	4804	4841	4850	4856	4860	4864						
G\$YES = 000010	785#	2630	4799	4800	4801	4802	4840	4842	4843	4844	4845	4846	4847
	4848	4849	4851	4852	4853	4854	4855	4857	4858	4859	4861	4862	4863

IFAUP	002444	1257#		
IFAU23	025232	3794#		
IFA10	014040	2392	2394#	
IFA11	014100	2412#		
IFA12	014244	2444#		
IFA121	014760	2540#		
IFA20	020750	3046#	3053	
IFA21	021520	3146	3149	3151#
IFA22	022322	3305#		
IFA23	022516	3345#		
IFA231	023410	3483#		
IFA232	024024	3582#		
IFA233	024262	3642#		
IFA234	024450	3681#		
IFA24	025360	3825#		
IFA241	025456	3849#		
IFA242	026232	3924#		
IFA244	030506	4325#		
IFA25	030720	4384#		
IFA251	031700	4497#		
IFA26	032354	4551#		
IFA40	032510	4592#		
IFB11	011710	1968	1972#	
IFB10	014024	2391#		
IFB12	014316	2453#		
IFB121	015010	2547#		
IFB13	017342	2752#		
IFB20	021036	3059#		
IFB21	021546	3152	3157#	
IFB22	022332	3307#		
IFB23	022606	3353	3360#	
IFB231	023504	3501#		
IFB232	024120	3599	3601#	
IFB233	024314	3643	3648	3650#
IFB242	026326	3925	3938#	
IFB244	030524	4331#		
IFB25	030770	4400#		
IFB251	031706	4499#		
IFB26	032370	4555#		
IFB40	032554	4606#		
IFC11	011724	1973	1976#	
IFC11	014166	2422	2424#	
IFC12	014420	2463	2467	2469#
IFC121	015116	2559	2562#	
IFC13	017350	2754#		
IFC20	021052	3062#		
IFC21	021570	3158	3162#	
IFC22	022360	3310	3312#	
IFC23	022714	3380	3382#	
IFC231	023526	3493	3496	3507#
IFC232	024136	3605#		
IFC233	024322	3652#		
IFC234	024602	3685	3701#	
IFC241	025560	3866#		
IFC242	026266	3930#		
IFC244	030536	4332	4335#	

IFC25	031030	4401	4409#		
IFC251	031730	4500	4504#		
IFC40	032614	4607	4615#		
IFD11	011744	1977	1981#		
IFD12	014430	2470	2472#		
IFD121	015156	2567	2570#		
IFD21	021622	3169#			
IFD22	022402	3306	3317#		
IFD23	022742	3385	3388#		
IFD231	023544	3512#			
IFD232	024146	3606	3608#		
IFD233	024344	3651	3655	3657#	
IFD234	024612	3703#			
IFD241	025570	3868#			
IFD242	026444	3953	3956#		
IFD244	030544	4337#			
IFD25	031156	4429#			
IFD251	031740	4506#			
IFE11	011764	1971	1975	1985#	
IFE12	014444	2475#			
IFE121	015164	2572#			
IFE21	022136	3220#			
IFE22	022430	3320	3322#		
IFE23	023064	3406	3408	3410#	
IFE232	024170	3609	3613#		
IFE233	024302	3647#			
IFE234	024744	3702	3721#		
IFE241	025604	3867	3869	3871#	
IFE242	026460	3957	3959#		
IFE244	030564	4338	4341#		
IFE25	031074	4418#			
IFE251	032016	4505	4519#		
IFF11	011772	1987#			
IFF12	014502	2481#			
IFF121	015174	2574#			
IFF20	021122	3071#			
IFF21	021510	3148#			
IFF23	023072	3412#			
IFF231	023552	3514#			
IFF232	024064	3593#			
IFF241	025614	3873#			
IFF242	026722	3997	4000#		
IFF25	031060	4415#			
IFF251	032026	4521#			
IFG11	011776	1989#			
IFG12	014576	2471	2473	2488	2494#
IFG20	021142	3075#			
IFG21	021714	3163	3165	3184#	
IFG23	023040	3383	3387	3403#	
IFG231	023422	3486#			
IFG232	024102	3583	3598#		
IFG242	026506	3940	3965#		
IFG251	031750	4508#			
IFH11	012040	1988	1998#		
IFH12	014400	2465#			
IFH121	015076	2558#			

IFH20	021210	3076	3084#						
IFH21	021752	3154	3156	3192#					
IFH23	023154	3409	3423#						
IFH231	023432	3488#							
IFH232	023755	3570#							
IFH234	024474	3686#							
IFH242	026536	3966	3971#						
IFH251	032036	4523#							
IFI11	012104	1986	2007#						
IFI121	015136	2563	2566#						
IFI20	021064	3063	3065#						
IFI21	021774	3193	3195	3197#					
IFI23	022552	3350	3352#						
IFI231	023346	3470#							
IFI241	025462	3851#							
IFI242	026516	3967#							
IFJ11	012114	2010#							
IFJ21	022030	3198	3203#						
IFJ23	022706	3379#							
IFJ241	025762	3887#							
IFJ242	026546	3968	3973#						
IFK11	012124	2013#							
IFK20	020700	3036#							
IFK21	021576	3164#							
IFK234	024532	3692#							
IFK241	025770	3889#							
IFK242	026574	3976	3979#						
IFL11	012164	2012	2021#						
IFL20	021246	3066	3092#						
IFL21	021476	3145#							
IFL241	026000	3891#							
IFL242	026656	3970	3978	3984	3990	3992#			
IFM20	021356	3109#							
IFM21	021654	3170	3172	3176#					
IFM241	025634	3877#							
IFM242	026622	3982	3985#						
IFN21	021664	3178#							
IFN242	026702	3962	3972	3993	3996#				
IFO21	022076	3211#							
IFO242	026742	4005#							
IFP242	026750	4007#							
IFQ242	026770	4006	4011#						
IFR242	027006	4008	4010	4012	4014#				
IFS21	027050	3964	3999	4002	4004	4018	4020	4022#	
IFU242	027024	4015	4017#						
IFV242	027032	4019#							
IFX242	026424	3952#							
IF00	013776	2369	2374#						
IF1211	015464	2635#							
IGATDP	012602	14#							
IG1211	015502	256	2638#						
IG243	027422	4077	4105#						
IHATDP	012616	2098	2100#						
IH1211	015554	2647#							
IH243	027456	4108	4114#						
IIATDP	012644	2106#							

PARAMETER CODING
CZRXC P11

29-MAR-82

MACY11 30(1046)
14:53

29-MAR-82 15:57 PAGE 82-15
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0120

L\$DU	013274	G	828	2210#					
L\$DUT	002072	G	828#						
L\$DVTY	002346	G	828	1208#					
L\$EF	002052	G	828#						
L\$ENVI	002044	G	828#						
L\$ERRT	002374	G	828	1251#					
L\$ETP	002102	G	828#						
L\$EXP1	002046	G	828#						
L\$EXP4	002064	G	828#						
L\$EXPS	002066	G	828#						
L\$HARD	035012	G	828	4797#					
L\$HIME	002120	G	828#						
L\$HPCP	002016	G	828#						
L\$HPTP	002022	G	828#						
L\$HW	002160	G	828	865#					
L\$ICP	002104	G	828#						
L\$INIT	011250	G	828	1889#					
L\$LADP	002026	G	828#						
L\$LAST	040454	G	828	4947#	4963				
L\$LOAD	002100	G	828#						
L\$LUN	002074	G	828#	1247*	2159*	2161*	2163*	2695*	
L\$MREV	002050	G	828#						
L\$NAME	002000	G	828#						
L\$PRIO	002042	G	828#						
L\$PROT	011242	G	828	1875#					
L\$PRT	002112	G	828#						
L\$REPP	002062	G	828#						
L\$REV	002010	G	828#						
L\$RPT	005140	G	828	1613#					
L\$SOFT	035136	G	828	4838#					
L\$SPC	002056	G	828#						
L\$SPCP	002020	G	828#						
L\$SPTP	002024	G	828#						
L\$STA	002030	G	828#						
L\$SW	002172	G	828	890#					
L\$TEST	002114	G	828#						
L\$TIML	002014	G	828#						
L\$UNIT	002012	G	828#	1907	1919				
L10000	002170		965	878#					
L10001	002220		890	911#					
L10002	004506		1348#						
L10003	004514		1368#						
L10004	005406		1649#						
L10006	011602		1933#						
L10007	012550		2070#						
L10010	012772		2130#						
L10011	013434		2230#						
L10012	013562		2270#						
L10013	035006		2379	4736#					
L10014	013664		2350#						
L10015	013750		2367#						
L10016	035060		4797	4804	4811#				
L10017	035400		4838	4864	4875#				
L10020	040460		4951#						
L10021	040474		4951	4957#					
L10022	040470		4951	4956#					

OSERRT=	000001	785#	818#	828						
OSGNSW=	000001	785#	818#	828						
OSPOIN=	000001	785#	818#	828						
OSETU=	000001	785#	818#	828	4947					
PAR	006026	1665*	1682*	1690*	1743	1746#				
PAT	017750	2746*	2782	2784*	2785	2864#				
PATCH	040246	4935#								
PAT125	017622	2799	2830#							
PAT333	017646	2800	2837#							
PG	017524	2804#	2807	2810						
PLOC	011604	1921*	1954#	1966						
PNT =	001000	982#								
POWERF =	000001	1064#	1898	2391						
PREPT1	006002	1667*	1684*	1694*	1717*	1743#				
PREPT2	006030	1660*	1687*	1755#						
PREPT3	006050	1726*	1765#							
PRESCK	020536	2984	2986#							
PRESTK	020642	2883*	2885*	2915	2923	2967	2986	3000*	3007#	
PRI =	002000	982#								
PRIDXX	006360	1626	1785#							
PRID01	006426	1785	1807#							
PRID02	006455	1786	1808#							
PRID03	006504	1787	1809#							
PRID04	006533	1788	1810#							
PRID05	006562	1789	1811#							
PRID06	006611	1790	1812#							
PRID07	006640	1791	1813#							
PRID08	006667	1792	1814#							
PRID09	006716	1793	1815#							
PRID10	006745	1794	1816#							
PRID11	006774	1795	1817#							
PRID12	007023	1796	1818#							
PRID13	007052	1797	1819#							
PRID14	007101	1798	1820#							
PRID15	007130	1799	1821#							
PRID16	007157	1800	1822#							
PRID17	007206	1801	1823#							
PRID18	007235	1802	1824#							
PRID19	007264	1803	1825#							
PRI00 =	000000	982#	1050#	3764						
PRI01 =	000040	982#	1049#							
PRI02 =	000100	982#	1048#							
PRI03 =	000140	982#	1047#							
PRI04 =	000200	982#	1046#							
PRI05 =	000240	982#	1045#							
PRI06 =	000300	982#	1044#							
PRI07 =	000340	982#	1043#	1929	1932	2153	3782			
PRNUM	005644	1624*	1634*	1644*	1680	1706#				
PRTBOS	004516	1370#	2497*	2715*						
PRTB1	004510	1366#	2166							
PRTB1S	004536	1367*	1373#	2652*	2713*	4533*				
PRTCTR	005646	1618*	1713#							
PRTDAT	005510	1628*	1639*	1648*	1679#					
PRTECD	002272	1115#	1257	1261*	1533*					
PRTERR	002404	1256#	2674*	4425*	4624*					
PRTHDR	005414	1617*	1621*	1631*	1642*	1658#				

SYSE6	032764	4636#	4652												
SYSE7	033010	4637#	4653												
SYSE8	033033	4638#	4654												
SYSE9	033072	4639#	4655												
S&LSYM=	010000	785#	878#	911#	1348#	1368#	1649#	1933#	2070#	2130#	2230#	2270#	2350#	2367#	
TARGET	020640	2542#	2612#	2630#	3055#	4736#	4811#	4875#							
		2884*	2886*	2917*	2919*	2925*	2927*	2940*	2943*	2947*	2955*	2959*	2962*	2969*	
		2972*	2974*	2988*	2991*	2997	2998	3000	3006#						
TBPRCT	022160	3160*	3164	3167*	3184	3186*	3189*	3214*	3216*	3229#					
THA234	024464	3682	3684#												
THA25	030756	4389	4396#												
THA40	032544	4597	4603#												
THB231	023514	3503#													
THC13	017366	2753	2757#												
THD23	022772	3392	3395#												
THE22	022440	3316	3324#												
THE234	024754	3723#													
THF231	023560	3487	3516#												
THF241	025624	3872	3875#												
TKTBPT	020634	2882*	2995*	2996	3004#										
TKTL	024242	3573	3584	3629#											
TKXX	010070	1645	1855#	4122											
TRACK	002254	1107#	3437*	3880	4118										
TRAP	013260	2153	2191#												
TRBIT =	000200 G	1054#	1540	2645	3688	3697	3705	3712	3716	3725					
TRKADR	025036	3433*	3437	3710	3739#										
TRKCNT	020636	2891*	2892*	2893*	2945*	2989*	3005#								
TRKDN	002260	1109#	3171	3173*	3617*										
TRKDNF	024230	3581*	3595*	3610*	3615*	3617	3624#								
TRKINC	024236	3414*	3582	3618*	3627#										
TRKSEQ	002202	896#	2748	2897											
TRKTBL	033554	2994	3587	4722#											
TRPMS1	013144	2158	2172#												
TSAVCT	022162	3201*	3207*	3209*	3211	3230#									
TSEC	033551	1260	4715#												
TST	021422	3058*	3059	3062	3071	3099	3103	3120#							
TSTCK	027626	3828*	4071	4151#	4200	4213	4238								
TSTEV	025410	3101*	3821	3823	3825	3828	3832#	3973	3979	3985	4000	4038			
TSTN	002176	894#	3056												
TSTPAT	002200	895#	2746												
TSTPTR	022152	3147*	3157	3159*	3166*	3180*	3182*	3187*	3190*	3199*	3205*	3210*	3217	3222*	
		3226#													
TSTSUT	022502	3309*	3311*	3312	3319*	3321*	3322	3334#							
TSTWD	022156	3058	3169	3176	3178	3219*	3228#								
TTRK	033550	1260	4714#												
TSARGC=	000004	828#	1256#	1259#	1260#	1288#	1370#	1373#	1743#	1755#	1765#	1926#	2035#	2038#	
		2220#	2228#	3880#	3881#										
TS&CODE=	001004	2630#	4799#	4800#	4801#	4802#	4804#	4840#	4841#	4842#	4843#	4844#	4845#	4846#	
		4847#	4848#	4849#	4850#	4851#	4852#	4853#	4854#	4855#	4856#	4857#	4858#	4859#	
		4860#	4861#	4862#	4863#	4864#									
TSERRN=	000000	785#													
TSEXCP=	000000	4799#	4800#	4801#	4802#	4843#	4844#	4845#	4846#	4857#	4858#	4861#	4862#		
TSFLAG=	000041	2379#	4804#	4864#											
TSFREE=	040504	4947	4963#												
TS&MAN=	000000	785#													
TS&HILI=	000032	4799#	4800#	4801#	4802#	4843#	4844#	4845#	4846#	4857#	4858#	4861#	4862#		

M\$POP	878#	911#	913#	1348#	1368#	1552#	1649#	1879#	1933#	2070#	2130#	2230#	2270#	2350#	2367#
M\$PRIN	2578#	2680#	3114#	4736#	4811#	4875#	4948#	1755#	1765#	1926#	2035#	2038#	2220#	2228#	3880#
M\$PUSH	1256#	1259#	1260#	1288#	1370#	1373#	1743#	1613#	1875#	1889#	2057#	2089#	2210#	2262#	2333#
M\$PUI	811#	865#	890#	975#	1347#	1366#	1605#	1755#	1765#	1926#	1929#	1932#	2035#	2038#	2153#
M\$PUT1	2335#	2356#	2542#	2612#	3055#	4797#	4838#	1755#	1765#	1926#	1929#	1932#	2035#	2038#	2153#
M\$RADI	1256#	1259#	1260#	1288#	1370#	1373#	1743#	1755#	1765#	1926#	1929#	1932#	2035#	2038#	2153#
M\$RNRO	2220#	2228#	3880#	3881#	1370#	1373#	1743#	4802#	4840#	4842#	4843#	4844#	4845#	4846#	4847#
M\$SETS	2630#	4799#	4800#	4801#	4802#	4840#	4842#	4843#	4844#	4845#	4846#	4847#	4848#	4849#	4851#
M\$SVC	4852#	4853#	4854#	4855#	4857#	4858#	4859#	4861#	4862#	4863#	1895#	1921#	811#	865#	890#
M\$TLAB	1895#	1921#	890#	975#	1347#	1366#	1605#	1613#	1875#	1889#	2057#	2089#	2210#	2262#	2333#
M\$TSTL	811#	865#	890#	975#	1347#	1366#	1605#	1613#	1875#	1889#	2057#	2089#	2210#	2262#	2333#
M\$WORD	2335#	2356#	2542#	2612#	3055#	4797#	4838#	1370#	1373#	1649#	1743#	1755#	1765#	1895#	1896#
M\$XFER	1248#	1256#	1259#	1260#	1288#	1348#	1368#	1370#	1373#	1649#	1743#	1755#	1765#	1895#	1896#
POINTE	1900#	1909#	1921#	1926#	1928#	1929#	1932#	1933#	2035#	2036#	2038#	2065#	2068#	2069#	2070#
PRINTB	2099#	2102#	2108#	2111#	2122#	2125#	2128#	2130#	2153#	2155#	2168#	2220#	2228#	2230#	2270#
PRINTF	2335#	2350#	2356#	2366#	2367#	2376#	2378#	2379#	2439#	2542#	2578#	2612#	2628#	2630#	2680#
PRINTS	2696#	2704#	3055#	3114#	3148#	3379#	3764#	3782#	3880#	3881#	4736#	4804#	4844#	4864#	4864#
PRINTX	1248#	1256#	1259#	1260#	1288#	1348#	1368#	1370#	1373#	1649#	1743#	1755#	1765#	1895#	1896#
READF	1900#	1909#	1921#	1926#	1928#	1929#	1932#	1933#	2035#	2036#	2038#	2065#	2068#	2069#	2070#
RFLAGS	2099#	2102#	2108#	2111#	2122#	2125#	2128#	2130#	2153#	2155#	2168#	2220#	2228#	2230#	2270#
SETPRI	2335#	2350#	2356#	2366#	2367#	2376#	2378#	2379#	2439#	2542#	2578#	2612#	2628#	2630#	2680#
SETVEC	2696#	2704#	3055#	3114#	3148#	3379#	3764#	3782#	3880#	3881#	4736#	4804#	4844#	4864#	4864#
SVC	1248#	1256#	1259#	1260#	1288#	1348#	1368#	1370#	1373#	1649#	1743#	1755#	1765#	1895#	1896#
XFER	1900#	1909#	1921#	1926#	1928#	1929#	1932#	1933#	2035#	2036#	2038#	2065#	2068#	2069#	2070#
XFERF	2099#	2102#	2108#	2111#	2122#	2125#	2128#	2130#	2153#	2155#	2168#	2220#	2228#	2230#	2270#
	2335#	2350#	2356#	2366#	2367#	2376#	2378#	2379#	2439#	2542#	2578#	2612#	2628#	2630#	2680#
	2696#	2704#	3055#	3114#	3148#	3379#	3764#	3782#	3880#	3881#	4736#	4804#	4844#	4864#	4864#
	828#	847#	2379#	2630#	4799#	4800#	4801#	4802#	4804#	4840#	4841#	4842#	4843#	4844#	4845#
	4846#	4847#	4848#	4849#	4850#	4851#	4852#	4853#	4854#	4855#	4856#	4857#	4858#	4859#	4860#
	4861#	4862#	4863#	4864#	4951#	4957#									
	4804#	4841#	4850#	4856#	4860#	4864#									
	818														
	1256	1370	1373	3880	3881										
	1926	2035	2038	2220	2228										
	1743	1755	1765												
	1259	1260	1288												
	1896	1900	1909												
	1895														
	3764	3782													
	1929	1932	2153												
	784#	785													
	2379#	4804#	4864#												
	4841	4850	4856	4860											

. ABS. 040504 000

ERRORS DETECTED: 0

CZRXDC,CZRXDC/CRF=SVC.SML/ML,CZRXDC.P11
 RUN-TIME: 22 22 3 SECONDS
 RUN-TIME RATIO: 379/48=7.8

PARAMETER CODING MACY11 30(1046) 29-MAR-82 15:57 PAGE 83-3
CZRxdc.P11 29-MAR-82 14:53 CROSS REFERENCE TABLE -- MACRO NAMES

C 11

SEQ 0132

CORE USED: 22K (44 PAGES)