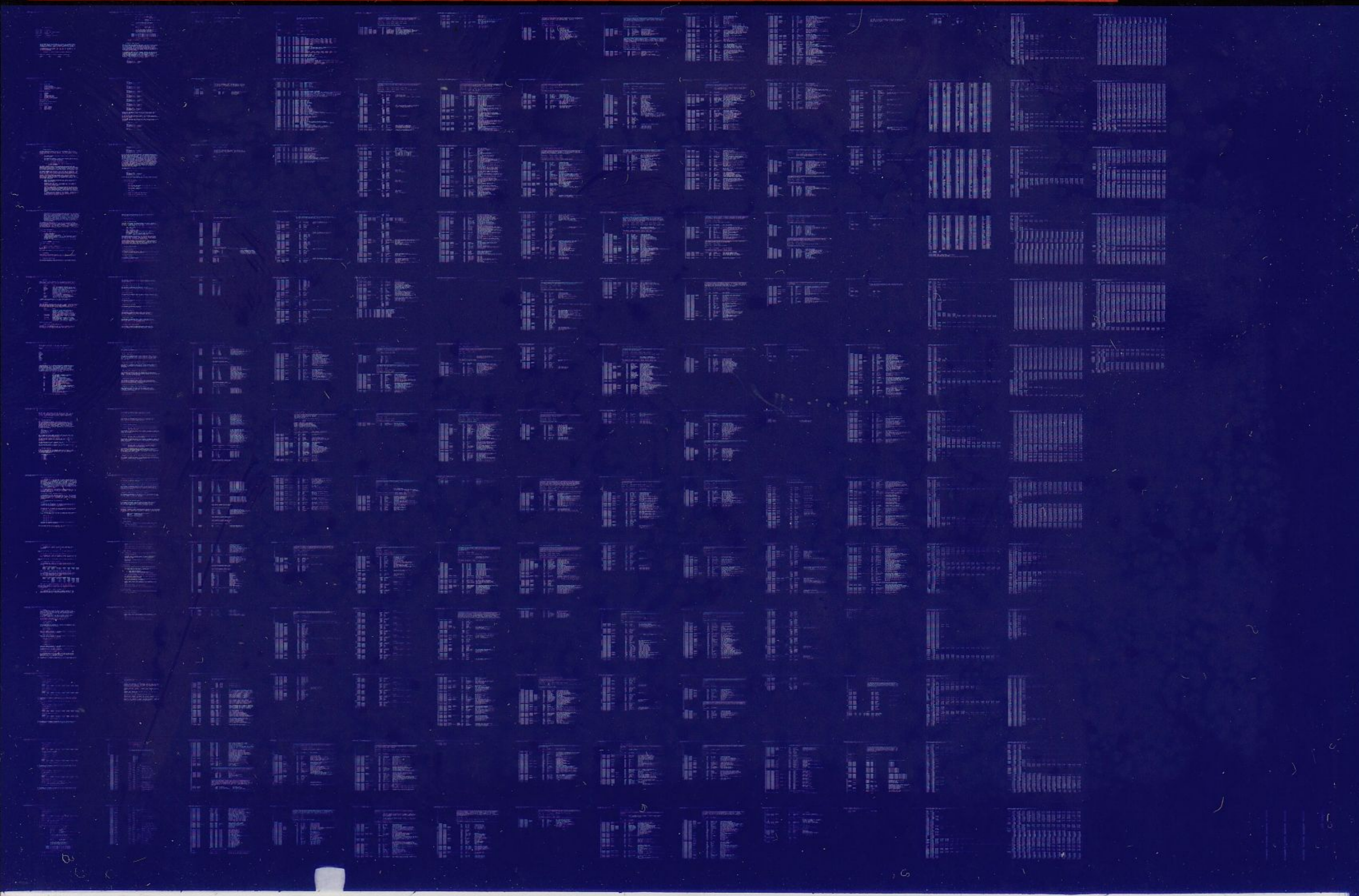


RP07

RP07 FMTR/SCANNER
CZAJKBO

COPYRIGHT (c) 1983
AH-F957B-MC
FICHE 1 OF 1

APR 1984
digital
Made In USA



.REM @

IDENTIFICATION

PRODUCT CODE: AC-F956B-MC
PRODUCT NAME: CZRJKBO RP07 FORMAT/SCANNER
PRODUCT DATE: DECEMBER 1, 1983
MAINTAINER: CX DIAGNOSTIC ENGINEERING
AUTHOR: MIKE LEAVITT

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) 1983 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL
DEC

PDP
DECUS

UNIBUS
DECTAPE

MASSBUS

@

.REM @

TABLE OF CONTENTS

- 1.0 GENERAL INFORMATION
 - 1.1 PROGRAM ABSTRACT
 - 1.2 SYSTEM REQUIREMENTS
 - 1.3 RELATED DOCUMENTS AND STANDARDS
 - 1.4 DIAGNOSTIC HIERARCHY PREREQUISITES
 - 1.5 ASSUMPTIONS
- 2.0 OPERATING INSTRUCTIONS
 - 2.1 COMMANDS
 - 2.2 SWITCHES
 - 2.3 FLAGS
 - 2.4 HARDWARE QUESTIONS
 - 2.5 SOFTWARE QUESTIONS
 - 2.6 EXTENDED P-TABLE DIALOGUE
 - 2.7 QUICK STARTUP PROCEDURE
- 3.0 ERROR INFORMATION
- 4.0 PERFORMANCE AND PROGRESS REPORTS
- 5.0 DEVICE INFORMATION TABLES
- 6.0 PROGRAM PROCESSES
 - 6.1 FORMAT PROCESS
 - 6.2 VERIFY PROCESS
 - 6.3 SCAN PROCESS
 - 6.4 MODIFY PROCESS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

THE RP07 FORMATTER PROGRAM FULFILLS THE REQUIREMENTS FOR A 16 BIT MANUFACTURING AND FIELD FORMAT PROCESS. THE FORMAT PROCESS FOR RP07 DIFFERS SIGNIFICANTLY FROM PREVIOUS RPXX DEVICE TECHNOLOGY IN SEVERAL RESPECTS:

- A) THE FORMAT PROCESS USES MEDIA "DEFECT SKIPS" IN ADDITION TO BAD SECTOR FLAGGING.
- B) THE FORMATTER PROGRAM, A STAND ALONE PROGRAM, CAN BE USED IN THE FIELD, BUT NOT INDISCRIMINATELY. THIS PROGRAM IS INTENDED FOR USE IN 2 DIFFERENT ENVIRONMENTS:
 - 1. AT THE VENDOR'S MANUFACTURING SITE
 - 2. AT THE CUSTOMERS SITE, USED BY TRAINED DEC FIELD SERVICE PERSONNEL

THE FORMAT PROGRAM ACCEPTS TRACK DESCRIPTOR DATA WHICH MAPS THE ABSOLUTE CENTER OF A DEFECT, PERFORMS CALCULATIONS ON THIS INPUT DATA, THEN CREATES AN OUTPUT MAP OF THE TRACK DEFECTS, SHOWING THE LOCATION OF THE DEFECT SKIP RELATIVE TO PLACEMENT WITHIN A GIVEN SECTOR. THIS MAP IS THEN USED TO FORMAT OR VERIFY THE FORMAT FOR A SPECIFIED TRACK.

THE FORMAT PROGRAM ALSO CONTAINS A PACK SCAN MODE OF OPERATION. WHEN RUN IN THIS MODE, DEFECTS WHICH CAUSE DATA OR HEADER ERRORS WILL BE MAPPED AND ADDED TO THE EXISTING TRACK DESCRIPTOR RECORDS AS NECESSARY.

THE RP07 FORMATTER PROGRAM HAS SEVERAL USES, THE MAIN FUNCTION HOWEVER, IS TO CREATE THE FORMAT USED FOR THE PDP11 (16 BIT) PRODUCTS. THIS FUNCTION, INTENDED FOR USE AT THE MANUFACTURING SITE BY THE MANUFACTURER, AND IN THE FIELD BY THE TRAINED FIELD ENGINEER, CONTAINS THE FOLLOWING PROCESSES:

- A) GIVEN THAT THE TRACK DESCRIPTORS ARE ALREADY WRITTEN ON THE DRIVE, THE PROGRAM WILL MAP THE STARTING LOCATIONS OF ALL NEEDED DEFECT SKIPS.
- B) DEPENDENT UPON USER INPUT, THE PROGRAM WILL THEN FORMAT OR FORMAT VERIFY EITHER A SELECTED TRACK, PART OF A DRIVE'S SURFACE OR AN ENTIRE DRIVE.
- C) ASSUMING A HIGH DEGREE OF SOPHISTICATION ON THE PART OF THE USER, THE PROGRAM WILL ALLOW MANUAL ENTRY OF TRACK DESCRIPTOR INFORMATION, THEN REWRITE A GIVEN TRACK DESCRIPTOR, PER USER INPUT VIA THE CONSOLE KEYBOARD. IMPLIED HERE IS THE NEED TO REFORMAT THE SPECIFIED TRACK BASED ON THE NEW TRACK DESCRIPTOR INPUT.
IF RUNNING THE FIELD VERSION OF THIS PROGRAM, WARNING MESSAGES WILL BE GENERATED TO INSURE THAT THE USER HAS A CHANCE TO CHANGE FUNCTIONS AND AVOID A POSSIBLE CATASTROPHY, I.E., DESTRUCTION OF A SYSTEM DATA FILE.

- D) WHEN PERSISTENT DATA OR HEADER ERRORS ARE DETECTED, THE FORMATTER CAN BE RUN IN THE PACK-SCAN MODE OF OPERATION. THIS OPERATION, WHICH RUNS TO COMPLETION PRIOR TO EXECUTING ANY FORMAT FUNCTION, WRITES THREE DATA DATA PATTERNS (ALL 0'S, ALL 1'S AND WORST CASE) ON THE USER SELECTED AREA IN AN ATTEMPT TO ISOLATE ANY UNMAPPED MEDIA DEFECTS. WHEN DEFECTS ARE DETECTED, THE ASSOCIATED TRACK DESCRIPTOR RECORD IS UPDATED (IN MEMORY ONLY) TO INCLUDE THE NEW DEFECT(S). UPON COMPLETION OF THIS MODE OF OPERATION, THE FORMATTER PROGRAM MAY BE RUN TO MASK THE NEW DEFECT LIST ON THE AFFECTED MEDIA.

THIS PROGRAM HAS BEEN WRITTEN FOR USE WITH THE DIAGNOSTIC RUNTIME SERVICES SOFTWARE (SUPERVISOR). THESE SERVICES PROVIDE THE INTERFACE TO THE OPERATOR AND TO THE SOFTWARE ENVIRONMENT. THIS PROGRAM CAN BE USED WITH XXDP+, ACT, APT, AND SLIDE. FOR A COMPLETE DESCRIPTION OF THE RUNTIME SERVICES, REFER TO THE XXDP+ USER'S MANUAL. THERE IS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES IN SECTION 2 OF THIS DOCUMENT.

1.2 SYSTEM REQUIREMENTS

THIS PROGRAM, IN ORDER TO EXECUTE, WILL REQUIRE THE FOLLOWING HARDWARE:

1. AN XXDP+ LOAD MEDIUM.
2. A CONSOLE TERMINAL
3. A MINIMUM OF 28K OF MAIN MEMORY.
4. A PDP11 PROCESSOR EXCEPT AN LSI 11, AND APPROPRIATE MASSBUS CONTROLLER WHICH CONFORMS TO DEC STD 159, AND HAS A DATA TRANSFER RATE EQUIVALENT TO OR GREATER THAN 2.2 MBYTES/SEC.
5. AT LEAST ONE RP07.
6. OPTIONALLY, A LINE PRINTER MAY ALSO BE USED BY THE PROGRAM.

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USER'S MANUAL - CHQUS

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

ALL CPU, MEMORY AND TERMINAL DIAGNOSTICS MUST RUN SUCCESSFULLY TO COMPLETION.

RP07 FRONT END DIAGNOSTIC MUST RUN SUCCESSFULLY TO COMPLETION.

1.5 ASSUMPTIONS

PROPER USE OF THIS PROGRAM REQUIRES THAT THE USER BE FAMILIAR WITH THE RULES WHICH APPLY TO DEFECT SKIPPING, AS APPLIED TO THE RP07.

2.0 OPERATING INSTRUCTIONS

THIS SECTION CONTAINS A BRIEF DESCRIPTION OF THE RUNTIME SERVICES. FOR DETAILED INFORMATION, REFER TO THE XXDP+ USER'S MANUAL (CHQUS).

2.1 COMMANDS

THERE ARE ELEVEN LEGAL COMMANDS FOR THE DIAGNOSTIC RUNTIME SERVICES (SUPERVISOR). THIS SECTION LISTS THE COMMANDS AND GIVES A VERY BRIEF DESCRIPTION OF THEM. THE XXDP+ USER'S MANUAL HAS MORE DETAILS.

COMMAND	EFFECT
START	START THE DIAGNOSTIC FROM AN INITIAL STATE
RESTART	START THE DIAGNOSTIC WITHOUT INITIALIZING
CONTINUE	CONTINUE AT TEST THAT WAS INTERRUPTED (AFTER +C)
PROCEED	CONTINUE FROM AN ERROR HALT
EXIT	RETURN TO XXDP+ MONITOR (XXDP+ OPERATION ONLY!)
ADD	ACTIVATE A UNIT FOR TESTING (ALL UNITS ARE CONSIDERED TO BE ACTIVE AT START TIME)
DROP	DEACTIVATE A UNIT
PRINT	PRINT STATISTICAL INFORMATION (IF IMPLEMENTED BY THE DIAGNOSTIC - SECTION 4.0)
DISPLAY	TYPE A LIST OF ALL DEVICE INFORMATION
FLAGS	TYPE THE STATE OF ALL FLAGS (SEE SECTION 2.3)
ZFLAGS	CLEAR ALL FLAGS (SEE SECTION 2.3)

A COMMAND CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. SO YOU MAY, FOR EXAMPLE, TYPE "STA" INSTEAD OF "START".

2.2 SWITCHES

THERE ARE SEVERAL SWITCHES WHICH ARE USED TO MODIFY SUPERVISOR OPERATION. THESE SWITCHES ARE APPENDED TO THE LEGAL COMMANDS. ALL OF THE LEGAL SWITCHES ARE TABULATED BELOW WITH A BRIEF DESCRIPTION OF EACH. IN THE DESCRIPTIONS BELOW, A DECIMAL NUMBER IS DESIGNATED BY "DDDD".

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10. THIS LIST WILL CAUSE TESTS 1,5,7,8,9,10 TO BE RUN. ALL OTHER TESTS WILL NOT BE RUN.
/PASS:DDDD	EXECUTE DDDDD PASSES (DDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDD = 1 TO 64000)
/UNITS:LIST	TEST/ADD/DROP ONLY THOSE UNITS SPECIFIED IN THE LIST. LIST EXAMPLE - /UNITS:0:5:10-12 USE UNITS 0,5,10,11,12 (UNIT NUMBERS = 0-63)

EXAMPLE OF SWITCH USAGE:

START/TESTS:1-5/PASS:1000/EOP:100

THE EFFECT OF THIS COMMAND WILL BE: 1) TESTS 1 THROUGH 5 WILL BE EXECUTED, 2) ALL UNITS WILL TESTED 1000 TIMES AND 3) THE END OF PASS MESSAGES WILL BE PRINTED AFTER EACH 100 PASSES ONLY. A

SWITCH CAN BE RECOGNIZED BY THE FIRST THREE CHARACTERS. YOU MAY, FOR EXAMPLE, TYPE "/TES:1-5" INSTEAD OF "/TESTS:1-5".

BELOW IS A TABLE THAT SPECIFIES WHICH SWITCHES CAN BE USED BY EACH COMMAND.

	TESTS	PASS	FLAGS	EOP	UNITS
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

2.3 FLAGS

FLAGS ARE USED TO SET UP CERTAIN OPERATIONAL PARAMETERS SUCH AS LOOPING ON ERROR. ALL FLAGS ARE CLEARED AT STARTUP AND REMAIN CLEARED UNTIL EXPLICITLY SET USING THE FLAGS SWITCH. FLAGS ARE ALSO CLEARED AFTER A START COMMAND UNLESS SET USING THE FLAG SWITCH. THE ZFLAGS COMMAND MAY ALSO BE USED TO CLEAR ALL FLAGS. WITH THE EXCEPTION OF THE START AND ZFLAGS COMMANDS, NO COMMANDS AFFECT THE STATE OF THE FLAGS; THEY REMAIN SET OR CLEARED AS SPECIFIED BY THE LAST FLAG SWITCH.

FLAG	EFFECT
HOE	HALT ON ERROR - CONTROL IS RETURNED TO RUNTIME SERVICES COMMAND MODE
LOE	LOOP ON ERROR
IER*	INHIBIT ALL ERROR REPORTS
IBR*	INHIBIT ALL ERROR REPORTS EXCEPT FIRST LEVEL (FIRST LEVEL CONTAINS ERROR TYPE, NUMBER, PC, TEST AND UNIT)
IXR*	INHIBIT EXTENDED ERROR REPORTS (THOSE CALLED BY PRINTX MACRO'S)
PRI	DIRECT MESSAGES TO LINE PRINTER
PNT	PRINT TEST NUMBER AS TEST EXECUTES
BOE	"BELL" ON ERROR
UAM	UNATTENDED MODE (NO MANUAL INTERVENTION)
ISR	INHIBIT STATISTICAL REPORTS (DOES NOT APPLY TO DIAGNOSTICS WHICH DO NOT SUPPORT STATISTICAL REPORTING)
IDR	INHIBIT PROGRAM DROPPING OF UNITS
ADR	EXECUTE AUTODROP CODE
LOT	LOOP ON TEST
EVL	EXECUTE EVALUATION (ON DIAGNOSTICS WHICH HAVE EVALUATION SUPPORT)

* ERROR MESSAGES ARE DESCRIBED IN SECTION 3.1

SEE THE XXDP+ USER'S MANUAL FOR MORE DETAILS ON FLAGS. YOU MAY SPECIFY MORE THAN ONE FLAG WITH THE FLAG SWITCH. FOR EXAMPLE, TO CAUSE THE PROGRAM TO LOOP ON ERROR, INHIBIT ERROR REPORTS AND TYPE A "BELL" ON ERROR, YOU MAY USE THE FOLLOWING STRING:

/FLAGS:LOE:IER:BOE

2.4 HARDWARE QUESTIONS

WHEN A DIAGNOSTIC IS STARTED, THE RUNTIME SERVICES WILL PROMPT THE USER FOR HARDWARE INFORMATION BY TYPING "CHANGE HW (L) ? " YOU MUST ANSWER "Y" AFTER A START COMMAND UNLESS THE HARDWARE INFORMATION HAS BEEN "PRELOADED" USING THE SETUP UTILITY (SEE CHAPTER 6 OF THE XXDP+ USER'S MANUAL). WHEN YOU ANSWER THIS QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING QUESTIONS FOR EACH UNIT.

UNIT 0
RPCS1 ADRS (0) 176700 ?
VECTOR ADRS (0) 254 ?
BR LEVEL (0) 5 ?
DRIVE # (0) 0 ?

THE 1ST QUESTION "RPCS1 ADRS" REQUIRES THAT THE USER INPUT THE RPCS1 ADDRESS OF THE CONTROLLER WHICH IS CONNECTED TO THE DRIVE UNDER TEST. DEFAULT IS 176700 (OCTAL).

THE 2ND QUESTION "VECTOR ADRS" REQUIRES THE USER TO INPUT THE INTERRUPT VECTOR ADDRESS OF THE RHXX CONTROLLER. DEFAULT IS 254 (OCTAL).

THE 3RD QUESTION "BR LEVEL" REQUIRES THE USER TO INPUT THE CONTROLLER INTERRUPT PRIORITY LEVEL. DEFAULT IS LEVEL 5.

THE 4TH QUESTION "DRIVE #" REQUIRES THE USER TO SPECIFY THE DRIVE NUMBER OF THE DRIVE TO BE TESTED. DEFAULT IS 0 (OCTAL).

2.5 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A START OR RESTART COMMAND, THE FOLLOWING PROMPT ASK YOU TO ENTER AN OPTION. THE SOFTWARE QUESTIONS AND THE DEFAULT VALUES ARE DESCRIBED IN THE NEXT PARAGRAPH(S).

"AVAILABLE OPTIONS FOLLOW:
0 =FORMAT
1 =VERIFY
2 =SCAN
3 =LIST
4 =MODIFY
5 =WRITE FE-2

ENTER OPTION (6=HELP) (D) 0 ? "

A '0' RESPONSE WILL ALLOW THE USER TO FORMAT HEADERS AND DATA ON THE DISK PACK. A '1' RESPONSE WILL ALLOW THE USER TO VERIFY TD'S AND HEADERS. A '2' RESPONSE WILL ALLOW THE USER TO SCAN THE DISK PACK FOR NEW DEFECTS AND RECORD THEM, IF DESIRED. A '3' RESPONSE WILL ALLOW THE USER TO LIST THE TRACKS WHICH HAVE DEFECTS AND LIST THE HEADER INFORMATION OF THE DEFECTIVE SECTORS, IF DESIRED. A '4' RESPONSE WILL ALLOW THE USER TO MODIFY THE TRACK DESCRIPTOR. A '5' RESPONSE WILL ALLOW THE USER TO WRITE THE SECOND FE CYLINDER ONLY. A '6' RESPONSE WILL PRINT A LIST OF VALID OPTIONS FOR THE FORMATTER PROGRAM (AS ABOVE).

IF THE FIELD VERSION OF THIS PROGRAM IS BEING RUN AND A '0', '2' OR '4' RESPONSE TO THE PREVIOUS OPTION PROMPT IS RECEIVED, THE FOLLOWING QUESTION WILL BE ASKED. OTHERWISE, THE NEXT TWO QUESTIONS ARE SKIPPED.

"DO YOU WANT TO WRITE ANYWHERE ON MEDIA (L) N ?"

IF THE RESPONSE TO THE PREVIOUS QUESTION IS 'N', THE FOLLOWING QUESTION WILL BE SKIPPED AND PROGRAM WILL PROCEED TO WRITE ONLY ON THE FE CYLINDER. A 'Y' RESPONSE WILL PRINT THE FOLLOWING WARNING MESSAGE TO THE OUTPUT DEVICE AND ASK THE FOLLOWING QUESTION.

" ! CUSTOMER DATA WILL BE OVERWRITTEN !

CONTINUE (L) ? "

THE USER MAY OPT TO RESTRICT THE PROGRAM TO WRITING ON THE FE CYLINDER ONLY BY ANSWERING 'N' TO THE QUESTION OR CONTINUE TO WRITE ON THE ENTIRE MEDIA BY ANSWERING 'Y'.

THE USER MAY OPT TO CHANGE THE DRIVE PARAMETERS IN CERTAIN MODES OF OPERATION. YOU MAY DO SO BY ANSWERING THE FOLLOWING PROMPT.

"CHANGE DRIVE PARAMETERS (L) N ? "

A 'Y' RESPONSE WILL ALLOW THE USER TO CHANGE THE DRIVE PARAMETERS, WHILE A 'N' RESPONSE WILL ASK THE NEXT APPROPRIATE QUESTION.

"MIN CYL (D) 0 ? "

"MAX CYL (D) 630 ? "

"MIN TRK (D) 0 ? "

"MAX TRK (D) 31 ? "

THE USER, BY STATING THE DESIRED PARAMETERS, MAY INCREASE OR DECREASE THE SCOPE OF OPERATION.

WHEN THE FORMAT OPTION (0) IS SELECTED, THE VERIFICATION AFTER A FORMAT MAY BE DESELECTED VIA THE FOLLOWING QUESTION.

"INHIBIT WRITE CHECK (L) N ? "

A 'Y' RESPONSE WILL INHIBIT THE WRITE-CHECK OPERATION. WHILE, A 'N' RESPONSE WILL TERMINATE CONSOLE DIALOGUE AND COMMENCE FORMATTING.

WHEN THE SCAN OPTION (2) IS SELECTED, THE FOLLOWING QUESTION WILL BE ASKED.

"DO YOU WANT TO UPDATE TD(S) WITH NEW DEFECT(S) (L) N ? "

A 'Y' RESPONSE WILL CAUSE THE PROGRAM TO FORMAT THE DISK WITH THE NEWLY FOUND DEFECTS AFTER THE COMPLETION OF THE CURRENT PACK SCAN.

WHEN THE LIST OPTION (3) IS SELECTED, THE FOLLOWING QUESTIONS WILL BE ASKED.

"LIST HEADER INFO IN DEFECT SECTOR(S) (L) N ? "

A 'N' RESPONSE WILL CAUSE THE FOLLOWING MESSAGE AND DATA TO BE PRINTED TO THE OUTPUT DEVICE OF A TRACK THAT HAS A DEFECT IN IT'S TD;

TD(S) WITH DEFECTS FOLLOW:

WORD#1 (CYL)	WORD#2 (TRK)(SEC)	WORD#3	WORD#4	WORD#5	WORD#6
031122 (594.)	004777 (9.)(377)	5415.	398.	140000	140000
071126 (598.)	004777 (9.)(377)	118.	4891.	108.	140000

NOTE: SEE TD & HEADER DEFINITIONS AT END OF THIS SECTION

A 'Y' RESPONSE WILL CAUSE THE FOLLOWING MESSAGE AND DATA TO BE PRINTED TO THE OUTPUT DEVICE OF A TRACK THAT HAS A DEFECT IN IT'S TD. THE ADDED HEADER INFO IS THE HEADER INFORMATION OF THE SECTOR IN WHICH THE DEFECT WAS PLACED;

TD(S) WITH DEFECTS FOLLOW (INCLUDING HEADER INFO):

WORD#1 (CYL)	WORD#2 (TRK)(SEC)	WORD#3	WORD#4	WORD#5	WORD#6
031122 (594.)	004777 (9.)(377)	5415.	398.	140000	140000
	004410 (8.)	176.	140000	140000	140000
	004440 (32.)	221.	140000	140000	140000
071126 (598.)	004777 (9.)(377)	118.	4891.	108.	140000
	004407 (7.)	52.	108.	140000	140000

NOTE: SEE TD & HEADER DEFINITIONS AT END OF THIS SECTION

WHEN THE MODIFY OPTION (4) IS SELECTED, THE FOLLOWING QUESTION WILL BE ASKED TO DETERMINE THE MEANS BY WHICH THE USER WANTS TO MODIFY THE TRACK DESCRIPTOR.

"MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 0 ? "

A '0' RESPONSE WILL ALLOW THE USER TO INSERT A DEFECT BY THE WORD COUNT POSITION IN A PARTICULAR SECTOR ON THE DISK. THIS METHOD SHOULD BE USED WHEN A DEFECT OCCURS WITHIN THE DATA AREA.

OF A SECTOR.

A '1' RESPONSE WILL ALLOW THE USER TO INSERT A DEFECT BY THE NUMBER OF WORDS FROM INDEX PULSE ON A PARTICULAR TRACK ADDRESS. THIS METHOD SHOULD BE USED WHEN THE TRACK DESCRIPTOR AND/OR HEADER 0 IS TO BE MOVED, DO TO A DEFECT WHICH OCCURED IN EITHER OR BOTH OF THESE AREAS.

A '2' RESPONSE WILL ALLOW THE USER TO CHANGE A DEFECT DIRECTLY IN THE TRACK DESCRIPTOR. THIS METHOD SHOULD BE USED WHEN RESTORING A TD BACK TO SOME ALREADY KNOWN CONDITION.

THE FOLLOWING 2 QUESTIONS WILL BE ASKED TO DETERMINE THE DISK ADDRESS WHICH IS TO BE MODIFIED BY THE USER.

"CYLINDER ADDRESS (D) 0 ? "

"TRACK ADDRESS (D) 0 ? "

AFTER DETERMINING THE DESIRED DISK ADDRESS TO BE MODIFIED, ONE OF FOLLOWING SETS OF QUESTIONS WILL BE ASKED, DEPENDING ON WHICH METHOD OF MODIFICATION WAS CHOSEN.

"SECTOR ADDRESS (D) ? "

"WORD IN SECTOR (D) ? "

AFTER THE CORRECT RESPONSE TO THE ABOVE PROMPTS, THE DIALOGUE WILL TERMINATE AND MODIFYING WILL COMMENCE.

"WORDS FROM INDEX (D) ? "

AFTER THE CORRECT RESPONSE TO THE ABOVE PROMPTS, THE DIALOGUE WILL TERMINATE AND MODIFYING WILL COMMENCE.

"TYPE <CR> TO INPUT (0-140000) DATA;
TD WORD#3 (D) 0 ?

TD WORD#4 (D) 0 ?

TD WORD#5 (D) 0 ?

TD WORD#6 (D) 0 ? "

AFTER THE CORRECT RESPONSE TO THE ABOVE PROMPTS, THE DIALOGUE WILL TERMINATE AND MODIFYING WILL COMMENCE.

ON COMPLETION OF THE MODIFY OPERATION, THE USER WILL BE PROMPTED WITH THE FOLLOWING QUESTION;

"DO YOU WANT TO MODIFY ANYMORE TD'S (L) N ? "

A 'N' RESPONSE WILL ALLOW THE USER TO CONTINUE TO NEXT DEVICE ON-LINE OR TO RETURN TO THE DRS PROMPT IF ONLY ONE DEVICE WAS BEING MODIFIED. A 'Y' RESPONSE WILL ALLOW THE USER TO MODIFY ANOTHER TD WITHOUT RESTARTING THE PROGRAM.

THE FOLLOWING IS AN EXAMPLE OF MODIFIING THE TD BY USING MODIFY OPTION '0' (WORDS MODE), TO INSERT A DEFECT WHICH IS IN THE DATA AREA OF A

PARTICULAR DISK ADDRESS;

EXAMPLE 1)

MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 0 ? <CR>

CYLINDER ADDRESS (D) 0 ? 598 <CR>

TRACK ADDRESS (D) 0 ? 9 <CR>

CURRENT TD,
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 (598.) 004777 (9.)(377) 140000 140000 140000 140000

SECTOR ADDRESS (D) ? 7 <CR>

WORD IN SECTOR (D) ? 150 <CR>

CHANGED TD,
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 (598.) 004777 (9.)(377) 5009. 140000 140000 140000

DO YOU WANT TO MODIFY ANYMORE TD'S (L) N ? Y <CR>

THE FOLLOWING IS AN EXAMPLE OF MODIFIING THE TD BY USING MODIFY OPTION '1' (INDEX MODE), TO INSERT A DEFECT WHICH IS IN TD OF OF A PARTICULAR DISK ADDRESS;

NOTE: USE 118. (DECIMAL) TO MOVE A TD AND 163. (DECIMAL) TO MOVE HDR 0.

EXAMPLE 2)

MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 0 ? 1 <CR>

CYLINDER ADDRESS (D) 598 ? <CR>

TRACK ADDRESS (D) 9 ? <CR>

CURRENT TD,
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 (598.) 004777 (9.)(377) 5009. 140000 140000 140000

WORDS FROM INDEX (D) ? 118 <CR>

CHANGED TD,
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
071126 (598.) 004777 (9.)(377) 118. 4891. 140000 140000

DO YOU WANT TO MODIFY ANYMORE TD'S (L) Y ? <CR>

THE FOLLOWING IS AN EXAMPLE OF MODIFIING THE TD BY USING MODIFY OPTION '2' (CHANGE MODE), TO INSERT A DEFECT ON A PARTICULAR DISK ADDRESS. IN THIS EXAMPLE, A STACKED DEFECT WILL BE ADDED TO THE LAST DEFECT;

EXAMPLE 3)

```
MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 1 ? 2<CR>
CYLINDER ADDRESS (D) 598 ? <CR>
TRACK ADDRESS (D) 9 ? <CR>
CURRENT TD.
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 (598.) 004777 (9.)(377) 118. 4891. 140000 140000
TYPE <CR> TO INPUT (0=140000) DATA;
TD WORD#3 (D) 0 ? 118<CR>
TD WORD#4 (D) 0 ? 4891<CR>
TD WORD#5 (D) 0 ? 108<CR>
TD WORD#6 (D) 0 ? <CR>
CHANGED TD.
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
071126 (598.) 004777 (9.)(377) 118. 4891. 108. 140000
DO YOU WANT TO MODIFY ANYMORE TD'S (L) Y ? <CR>
```

THE FOLLOWING IS AN EXAMPLE OF MODIFYING THE TD BY USING MODIFY OPTION '0' (WORDS MODE), TO INSERT A DEFECT ON A PARTICULAR DISK ADDRESS, WHICH ALREADY HAS 4 DEFECTS IN THE TD.

EXAMPLE 4)

```
MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 2 ? 0<CR>
CYLINDER ADDRESS (D) 598 ? 0<CR>
TRACK ADDRESS (D) 9 ? 6<CR>
CURRENT TD.
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 (0.) 004777 (6.)(377) 11069. 1425. 2641. 329.
SECTOR ADDRESS (D) ? 28<CR>
WORD IN SECTOR (D) ? (ANY VALID NUMBER)<CR>
WARNING ** CYL 0., TRK 6. HAS 4 DEFECTS IN TRACK DESCRIPTOR!
BAD SECTOR FILE WILL BE UPDATED
DO YOU WANT TO MODIFY ANYMORE TD'S (L) Y ? <CR>
```

THE FOLLOWING IS AN EXAMPLE OF MODIFYING THE TD BY USING MODIFY OPTION '1' (INDEX MODE), TO INSERT A DEFECT ON A PARTICULAR DISK ADDRESS, WHICH ALREADY HAS 4 DEFECTS IN THE TD.

EXAMPLE 5)

```
MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE) (D) 0 ? 1<CR>
CYLINDER ADDRESS (D) 0 ? <CR>
TRACK ADDRESS (D) 6 ? <CR>

CURRENT TD.
WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6
031126 ( 0.) 004777 ( 6.)(377) 11069. 1425. 2641. 329.

WORDS FROM INDEX (D) ? (ANY VALID NUMBER)<CR>
WARNING ** CYL 0., TRK 6. HAS 4 DEFECTS IN TRACK DESCRIPTOR!
          INPUT DATA REJECTED

DO YOU WANT TO MODIFY ANYMORE TD'S (L) Y ? N<CR>
```

THE FOLLOWING ARE THE DEFINITIONS FOR THE TD WORDS 1 THRU 6;

WHERE, 'WORD#1' IS THE 1ST WORD OF THE TD (IN OCTAL),

BITS 0-9	CYLINDER ADDR
BITS 10-11	ALWAYS 0
BIT 12	ALWAYS 1
BIT 13	TD (IDENTIFIES THIS IS TD RECORD)
BIT 14	TD MOVED
BIT 15	HDR 0 MOVED

'CYL' IS THE CYLINDER ADDR (IN DECIMAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 1.

'WORD#2' IS THE 2ND WORD OF THE TD (IN OCTAL),

BITS 0-7	SECTOR ADDR (ALWAYS ONES)
BITS 8-15	TRACK ADDR

'TRK' IS THE TRACK ADDR (IN DECIMAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 2.

'SEC' IS THE SECTOR ADDR (IN OCTAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 2.

'WORD#3' IS THE 3RD WORD OF THE TD (IN DECIMAL *),
'WORD#4' IS THE 4TH WORD OF THE TD (IN DECIMAL *),
'WORD#5' IS THE 5TH WORD OF THE TD (IN DECIMAL *),
'WORD#6' IS THE 6TH WORD OF THE TD (IN DECIMAL *).

SKIP DEFECT DISPLACEMENTS.

* IF SKIP DISPLACEMENT IS A 'NULL', THE OUTPUT WILL BE 140000
OCTAL.

THE FOLLOWING ARE THE DEFINITIONS FOR THE HEADER WORDS 1 THRU 6;

WHERE, 'WORD#1' IS THE 1ST WORD OF THE HDR (IN OCTAL),

BITS 0-11	CYLINDER ADDR
BIT 12	FMT BIT 1=16 BIT FMT 0=18 BIT FMT
BIT 13	ALWAYS 0
BIT 14-15	USED FOR BAD SECTOR FLAGGING 11=GOOD

'CYL' IS THE CYLINDER ADDR (IN DECIMAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 1,

'WORD#2' IS THE 2ND WORD OF THE TD (IN OCTAL),

BITS 0-7	SECTOR ADDR (0-49.)
BITS 8-15	TRACK ADDR (0-31.)

'TRK' IS THE TRACK ADDR (IN DECIMAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 2,

'SEC' IS THE SECTOR ADDR (IN OCTAL) OF THE TD,
WHICH WAS EXTRACTED FROM WORD 2,

'WORD#3' IS THE 3RD WORD OF THE TD (IN DECIMAL *),
'WORD#4' IS THE 4TH WORD OF THE TD (IN DECIMAL *),
'WORD#5' IS THE 5TH WORD OF THE TD (IN DECIMAL *),
'WORD#6' IS THE 6TH WORD OF THE TD (IN DECIMAL *),

SKIP DEFECT DISPLACEMENTS.

* IF SKIP DISPLACEMENT IS A 'NULL', THE OUTPUT WILL BE 140000
OCTAL.

2.6 EXTENDED P-TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES
IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST
WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH
UNIT TO BE TESTED. IF YOU HAVE A MULTIPLEXED DEVICE SUCH AS
A MASS STORAGE CONTROLLER WITH SEVERAL DRIVES OR A COMMUNICATION
DEVICE WITH SEVERAL LINES, THIS BECOMES TEDIOUS SINCE MOST OF
THE ANSWERS ARE REPETITIOUS.

TO ILLUSTRATE A MORE EFFICIENT METHOD, SUPPOSE YOU ARE TESTING
A FICTIONAL DEVICE, THE XY11. SUPPOSE THIS DEVICE CONSISTS OF
A CONTROL MODULE WITH EIGHT UNITS (SUB-DEVICES) ATTACHED TO IT.
THESE UNITS ARE DESCRIBED BY THE OCTAL NUMBERS 0 THROUGH 7. THERE
IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE
Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY
TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

* UNITS (0) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0<CR>
Q-FACTOR (0) 0 ? 1<CR>

UNIT 2
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 1<CR>
Q-FACTOR (0) 1 ? 0<CR>

UNIT 3
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 2<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 4
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 3<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 5
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 4<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 6
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 5<CR>
Q-FACTOR (0) 0 ? <CR>

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6<CR>
Q-FACTOR (0) 0 ? 1<CR>

UNIT 8
CSR ADDRESS (0) 160000<CR>
SUB-DEVICE # (0) ? 7<CR>
Q-FACTOR (0) 1 ? <CR>

NOTICE THAT THE DEFAULT VALUE FOR THE Q-FACTOR CHANGES WHEN A
NON-DEFAULT RESPONSE IS GIVEN. BE CAREFUL WHEN SPECIFYING
MULTIPLE UNITS!

AS YOU CAN SEE FROM THE ABOVE EXAMPLE, THE HARDWARE PARAMETERS
DO NOT VARY SIGNIFICANTLY FROM UNIT TO UNIT. THE PROCEDURE SHOWN IS
NOT VERY EFFICIENT.

THE RUNTIME SERVICES CAN TAKE MULTIPLE UNIT SPECIFICATIONS HOWEVER.
LET'S BUILD THE SAME TABLE USING THE MULTIPLE SPECIFICATION
FEATURE.

* UNITS (0) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0,1<CR>
Q-FACTOR (0) 0 ? 1,0<CR>

UNIT 3
CSR ADDRESS (0) ? 160000<CR>

SUB-DEVICE # (0) ? 2-5<CR>
Q-FACTOR (0) 0 ? 0<CR>

UNIT 7
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 6,7<CR>
Q-FACTOR (0) 0 ? 1<CR>

AS YOU CAN SEE IN THE ABOVE DIALOGUE, THE RUNTIME SERVICES WILL BUILD AS MANY ENTRIES AS IT CAN WITH THE INFORMATION GIVEN IN ANY ONE PASS THROUGH THE QUESTIONS. IN THE FIRST PASS, TWO ENTRIES ARE BUILT SINCE TWO SUB-DEVICES AND Q-FACTORS WERE SPECIFIED. THE SERVICES ASSUME THAT THE CSR ADDRESS IS 160000 FOR BOTH SINCE IT WAS SPECIFIED ONLY ONCE. IN THE SECOND PASS, FOUR ENTRIES WERE BUILT. THIS IS BECAUSE FOUR SUB-DEVICES WERE SPECIFIED. THE "-" CONSTRUCT TELLS THE RUNTIME SERVICES TO INCREMENT THE DATA FROM THE FIRST NUMBER TO THE SECOND. IN THIS CASE, SUB-DEVICES 2, 3, 4 AND 5 WERE SPECIFIED. (IF THE SUB-DEVICE WERE SPECIFIED BY ADDRESSES, THE INCREMENT WOULD BE BY 2 SINCE ADDRESSES MUST BE ON AN EVEN BOUNDARY.) THE CSR ADDRESSES AND Q-FACTORS FOR THE FOUR ENTRIES ARE ASSUMED TO BE 160000 AND 0 RESPECTIVELY SINCE THEY WERE ONLY SPECIFIED ONCE. THE LAST TWO UNITS ARE SPECIFIED IN THE THIRD PASS.

THE WHOLE PROCESS COULD HAVE BEEN ACCOMPLISHED IN ONE PASS AS SHOWN BELOW.

* UNITS (0) ? 8<CR>

UNIT 1
CSR ADDRESS (0) ? 160000<CR>
SUB-DEVICE # (0) ? 0-7<CR>
Q-FACTOR (0) 0 ? 0,1,0,...,1,1<CR>

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (COMMAS ENCLOSING A NULL FIELD) TELL THE RUNTIME SERVICES TO REPEAT THE LAST REPLY.

2.7 QUICK START-UP PROCEDURE (XXDP*)

TO START-UP THIS PROGRAM:

1. BOOT XXDP*
2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE IS A CLOCK) QUESTIONS
3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR BIC FILE FOR THIS PROGRAM
4. TYPE "START"
5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"
6. ANSWER ALL THE HARDWARE QUESTIONS
7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

WHEN YOU FOLLOW THIS PROCEDURE YOU WILL BE USING ONLY THE
DEFAULTS FOR FLAGS AND SOFTWARE PARAMETERS. THESE DEFAULTS
ARE DESCRIBED IN SECTIONS 2.3 AND 2.5.

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

THERE ARE THREE LEVELS OF ERROR MESSAGES THAT MAY BE ISSUED BY
A DIAGNOSTIC: GENERAL, BASIC AND EXTENDED. GENERAL ERROR MESSAGES
ARE ALWAYS PRINTED UNLESS THE "IER" FLAG IS SET (SECTION 2.3).
THE GENERAL ERROR MESSAGE IS OF THE FORM:

NAME TYPE NUMBER ON UNIT NUMBER TST NUMBER PC:XXXXXX
ERROR MESSAGE

WHERE: NAME = DIAGNOSTIC NAME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

BASIC ERROR MESSAGES ARE MESSAGES THAT CONTAIN SOME ADDITIONAL
INFORMATION ABOUT THE ERROR. THESE ARE ALWAYS PRINTED UNLESS
THE "IER" OR "IBR" FLAGS ARE SET (SECTION 2.3). THESE MESSAGES
ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

EXTENDED ERROR MESSAGES CONTAIN SUPPLEMENTARY ERROR INFORMATION
SUCH AS REGISTER CONTENTS OR GOOD/BAD DATA. THESE ARE ALWAYS
PRINTED UNLESS THE "IER", "IBR" OR "IXR" FLAGS ARE SET (SECTION 2.3).
THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR
MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

3.2 SPECIFIC ERROR MESSAGES

- TD ADDRESS INCORRECT -

THIS MESSAGE IS GENERATED WHEN A TRACK DESCRIPTOR IS READ AND FOUND
TO CONTAIN THE WRONG ADDRESS INFORMATION.

- TD FORMAT INCORRECT -

THIS MESSAGE IS GENERATED WHEN A TRACK DESCRIPTOR IS FOUND
TO CONTAIN AN INCORRECT FORMAT.

- COMPOSITE ERROR SET -

THIS MESSAGE IS GENERATED WHEN A COMPOSITE ERROR IS DETECTED.
COMPOSITE ERROR (RPDS:BIT 14) SETS WHEN A HARDWARE MALFUNCTION
HAS BEEN DETECTED.

- DRIVE HUNG, DRY NOT SET IN TIME -

THIS MESSAGE IS GENERATED WHEN RPDS:DRY (BIT 07)
DOES NOT SET WITHIN A GIVEN TIME PERIOD, IT INDICATES THAT
"GO" (RPCS1:BIT 0) DID NOT RESET IN TIME, THAT THE DRIVE IS HUNG.

- DRIVE WRITE LOCKED -

THIS MESSAGE IS GENERATED WHEN THE WRITE LOCK BIT (RPDS:BIT 11)
IS FOUND TO BE SET WHEN A WRITE OPERATION IS ABOUT TO BE PERFORMED.

- DRIVE OFFLINE -

THIS MESSAGE IS GENERATED WHEN MEDIUM-ON-LINE (RPDS:BIT12)
IS FOUND TO BE RESET.

- UNEXPECTED ATTN OCCURRED -

THIS MESSAGE IS GENERATED ANYTIME AN "ATTENTION" (RPDS: BIT 15)
IS RECEIVED WHEN IT IS NOT EXPECTED.

- WRITE CHECK ERROR SET -

THIS MESSAGE IS GENERATED WHEN A WRITE CHECK OPERATION IS
PERFORMED SUBSEQUENT TO A FORMAT OPERATION, AND THE
WRITE CHECK ERROR (RPCS2:BIT 14) IS FOUND TO BE ASSERTED.

- FORMAT VERIFY ERROR -

THIS MESSAGE IS GENERATED WHEN A VERIFY OPERATION FAILS TO
RECEIVE THE SAME DATA FROM THE DRIVE AS THE COMPUTED EXPECTED
DATA, RESULTING IN A DATA MISCOMPARE ERROR.

- DRY DIDN'T RESET WHEN EXPECTED -

THIS MESSAGE IS GENERATED WHEN DRIVE READY (RPDS:BIT 07)
FAILS TO RESET AFTER A COMMAND HAS BEEN ISSUED, INDICATING
THAT THE COMMAND MAY HAVE NOT BEEN EXECUTED BY THE DRIVE.

- TRANSFER ERROR SET -

THIS MESSAGE IS GENERATED WHEN A CONTROLLER ERROR "TRANSFER ERROR"
HAS BEEN DETECTED (RPCS1:BIT 14).

- SPECIAL CONDITION SET -

THIS MESSAGE IS GENERATED WHEN SPECIAL CONDITION (RPCS1; BIT 15)
IS FOUND TO BE ASSERTED WHEN IT IS NOT EXPECTED. SPECIAL CONDITION
CAN BE SET BY ANY DRIVE ON THE STRING AS WELL AS THE UNIT UNDER
TEST. IT CAN INDICATE A SUB-SYSTEM FAILURE OR A FAILURE OF THE
UNIT UNDER TEST.

- RANDOM HEADER ERRORS -CAN'T RESOLVE DEFECTS- -

THIS MESSAGE IS GENERATED WHEN TOO MANY RANDOM HEADER ERRORS ARE
DETECTED. IT INDICATES THE PRESENCE OF NOISE OR A MARGINALLY
OPERATIONAL RP07 WHICH MUST BE REPAIRED BEFORE THE SCAN OPERATION
CAN PROCEED.

- COULD NOT SUPPLY READABLE TD THRU RELOCATION -

THIS MESSAGE IS GENERATED WHEN THE SCANNER MODE OF OPERATION IS
UNABLE TO RESTORE A TRACK DESCRIPTOR RECORD THROUGH RELOCATION OF THAT
RECORD. THE TRACK IN QUESTION MUST THEN BE RETIRED VIA THE OPERATING
SYSTEM, OR THE HDA MUST BE REPLACED.

- UNRECOVERABLE ERROR DURING PACK SCAN -

THIS MESSAGE IS GENERATED WHEN, DURING A PACK SCAN OPERATION,
TOO MANY NON-DATA ERRORS ARE DETECTED DURING THE WRITE DATA OR
WRITE CHECK DATA OPERATIONS. EXCESSIVE FAILURES OF THIS TYPE USUALLY
INDICATE THE PRESENCE OF BROKEN HARDWARE.

- RANDOM WRT CHK ERRORS -CAN'T RESOLVE DEFECTS- -

THIS MESSAGE IS GENERATED WHEN TOO MANY RANDOM WRITE CHECK
ERRORS ARE DETECTED. IT INDICATES THE PRESENCE OF NOISE OR
A MARGINALLY OPERATIONAL RP07 WHICH MUST BE REPAIRED BEFORE THE
SCAN OPERATION CAN PROCEED.

- NON-EXISTENT DRIVE -

THIS MESSAGE IS GENERATED WHEN A USER SELECTS A DRIVE WHICH CAUSES NON EXISTENT DRIVE (RPCS2:BIT 11) TO SET.

- DRIVE NOT AN RP07 -

THIS MESSAGE IS GENERATED WHEN THE DRIVE TYPE REGISTER (RPDT) DOESN'T CONTAIN THE CORRECT IDENTIFICATION, INDICATING THAT A DRIVE OTHER THAN A RP07 HAS BEEN SELECTED.

3.3 SPECIFIC WARNING MESSAGES

- WARNING ** CYL XXX., TRK YY. HAS 4 DEFECTS IN TRACK DESCRIPTOR -
- INPUT DATA REJECTED -

THIS MESSAGE IS GENERATED WHEN A USER ATTEMPTS TO MANUALLY (USING THE 'INDEX' OPTION) ADD A DEFECT SKIP TO A TRACK DESCRIPTOR RECORD WHICH ALREADY HAS 4 DEFECTS.

- WARNING ** CYL XXX., TRK YY. HAS 4 DEFECTS IN TRACK DESCRIPTOR -
- BAD SECTOR FILE WILL BE UPDATED -

THIS MESSAGE IS GENERATED WHEN A USER ATTEMPTS TO MANUALLY (USING THE 'WORDS' OPTION) ADD A DEFECT SKIP TO A TRACK DESCRIPTOR RECORD WHICH ALREADY HAS 4 DEFECTS. THIS DEFECT IS THEN LOGGED IN THE BAD SECTOR FILE AS A DEFECT.

- WARNING ** CYL XXX., TRK YY. HAS 4 DEFECTS IN TRACK DESCRIPTOR -
- ANOTHER DEFECT WAS FOUND XXXX. WORDS FROM INDEX -
- BAD SECTOR FILE WILL BE UPDATED -

THIS MESSAGE IS GENERATED WHEN THE SCANNER ATTEMPTS TO ADD A DEFECT SKIP TO A TRACK DESCRIPTOR RECORD WHICH ALREADY HAS 4 DEFECTS. WHERE, XXXX IS THE NUMBER OF WORDS FROM THE INDEX PULSE TO THE POINT OF THE DEFECT JUST FOUND. (DECIMAL) THIS DEFECT IS THEN LOGGED IN THE BAD SECTOR FILE AS A DEFECT.

- WARNING ** CYL XXX., TRK YY. HAS 4 DEFECTS IN TRACK DESCRIPTOR -

THIS MESSAGE IS GENERATED WHEN A TD IS DETECTED WITH FOUR DEFECTS IN IT'S RECORD. WHERE, XXX IS THE CYLINDER ADDRESS AND YY IS THE TRACK ADDRESS OF THE TD JUST READ. (DECIMAL)

- WARNING ** BAD SECTOR FILE BUFFER IS FULL, NO MORE ENTRIES ALLOWED -
THIS MESSAGE IS GENERATED WHEN AN ATTEMPT IS MADE TO ADD A BAD SPOT
TO A FULL BAD SECTOR FILE (DEC144).

- WARNING ** BAD SECTOR FILE DATA IS CORRUPT, IT WILL BE INITIALIZED -
THIS MESSAGE IS GENERATED WHEN AN ATTEMPT TO READ THE BAD SECTOR FILE
TRACK, RESULTS IN AN UNRECOVERABLE READ ERROR OR CORRUPT DATA IS
DETECTED IN THE CONTENTS OF THE FILE. THE BSF WILL BE INITIALIZED ON
COMPLETION OF THE TEST.

- WARNING ** THERE ARE XX. TRACK(S) WITH 4 DEFECTS -
THIS MESSAGE IS GENERATED AT THE END OF A PASS, TO REMINDER THE USER
THAT SOME NUMBER OF TRACKS HAD FOUR DEFECTS. WHERE, XX IS THE NUMBER
OF TRACKS HAVING FOUR DEFECTS. (DECIMAL)

4.0 PERFORMANCE AND PROGRESS REPORTS

AT THE END OF EACH PASS, THE PASS COUNT IS GIVEN ALONG WITH THE
TOTAL NUMBER OF ERRORS REPORTED SINCE THE PROGRAM WAS STARTED.
THE "EOP" SWITCH CAN BE USED TO CONTROL HOW OFTEN THE END
OF PASS MESSAGE IS PRINTED. SECTION 2.2 DESCRIBES SWITCHES.

5.0 DEVICE INFORMATION TABLES

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 HARDWARE P-TABLES, AND IS USED AS A "TEMPLATE" FOR BUILDING THE
P-TABLES.

.WORD 176700 ;RPCS1 BASE REGISTER ADDRESS
.WORD 254 ;VECTOR ADDRESS
.WORD 240 ;BR LEVEL 5 DEVICE
.WORD 0 ;DRIVE NUMBER

6.0 PROGRAM PROCESSES

6.1 FORMAT PROCESS

A FORMAT PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS:

1. READ TRACK DESCRIPTOR
2. FORMAT TRACK (WRITE TRACK HEADERS & DATA ACCORDING TO THE

DEFECT INFO IN THE TRACK DESCRIPTOR & MFG BAD SECTOR FILE
INFO)

3. VERIFY TRACK FORMAT, WITH WRITE CHECK (IF ENABLED)
4. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.2 VERIFY PROCESS

A VERIFY PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

1. READ TRACK DESCRIPTOR
2. CONSTRUCT MAP OF EXPECTED FORMAT IN MEMORY BASED ON CONTENTS OF TRACK DESCRIPTOR JUST READ & THE MFG BAD SECTOR FILE INFO.
3. READ HEADERS (COMMAND MODIFIER BIT=1) AND CHECK AGAINST MEMORY MAP.
4. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.3 SCAN PROCESS

A SCAN PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

1. SCAN PACK TRACK BY TRACK BASIS, LOOKING FOR DEFECTS IN THE TD, HEADER AND DATA PORTIONS OF EACH SECTOR.
 - A. READ TRACK DESCRIPTOR
 - B. READ HEADERS (COMMAND MODIFIER BIT=1)
 - C. WRITE DATA & VERIFY WITH WRITE CHECK COMMAND

NOTE: THE SCAN PROCESS TAKES APPROX. 135 MINUTES PER DRIVE.

2. ANY PERSISTENT ERRORS WILL CAUSE THE TRACK DESCRIPTOR TO BE READ AND APPENDED IN MEMORY.
3. WRITE TRACK DESCRIPTOR (ACCORDING TO THE BUFFER CREATED BY THE PACK SCAN OPERATION. IF ENABLED)
4. FORMAT TRACK (WRITE TRACK HEADERS & DATA ACCORDING TO THE DEFECT INFO IN THE TRACK DESCRIPTOR & MFG BAD SECTOR FILE INFO)
5. VERIFY TRACK FORMAT, WITH WRITE CHECK
6. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.4 MODIFY PROCESS

A MODIFY PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

1. READ TRACK DESCRIPTOR

2. MODIFY TRACK DESCRIPTOR
3. WRITE TRACK DESCRIPTOR
4. FORMAT TRACK (WRITE TRACK HEADERS & DATA ACCORDING TO THE DEFECT INFO IN THE TRACK DESCRIPTOR & MFG BAD SECTOR FILE INFO)
5. VERIFY TRACK FORMAT, WITH WRITE CHECK
6. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

a

.REM 8

VERSION (CZRJK-A-0)

- 1. THIS VERSION IS THE STARTING POINT FOR CX DIAGNOSTIC SUPPORT OF THE RP07 DISK DRIVE.

VERSION (CZRJK-B-0)

- 1. CHANGED THE ROUTINES THAT DETERMINE WHEN TO MOVE THE TRACK DESCRIPTOR AND/OR HEADER 0. THE CHANGE WAS NEEDED BECAUSE OF A PARAMETER LIMIT DIFFERENCE BETWEEN THIS PROGRAM AND THE MANUFACTURING SCANNER PROGRAM.
- 2. CHANGED ROUTINES 'TDSCAN', 'HDSCAN' AND 'DASCAN' TO DO 10. ITERATIONS AFTER DETECTING A MEDIA RELATED ERROR DURING SCAN MODE.
- 3. CHANGED THE HEADER SCAN ROUTINE 'HDSCAN', TO LOOK FOR 'OPI' ERRORS DURING SCAN MODE.
- 4. CORRECTED A SHIFT RIGHT PROBLEM IN ROUTINE 'TOLER'.
- 5. MODIFIED THE PROGRAM TO FORMAT THE MF BIT CORRECTLY INTO A HEADER THAT HAS BEEN FOUND IN THE MFG BAD SECTOR FILE.
- 6. MODIFIED THE PROGRAM TO ALLOW THE USER TO FORMAT, SCAN OR MODIFY TD'S ON THE FE CYLINDER, WITHOUT HAVING TO CHANGE THE MIN/MAX CYLINDER LIMITS IN THE DRIVE PARAMETERS.

8

HISTORY

1
2
136
138
164
166 000000
167 002000
169
171
172
173
174
175
177
194
198 002000
002000 103
002001 132
002002 122
002003 112
002004 113
002005 000
002006 000
002007 000
002010
002010 102
002011
002011 060
002012
002012 000001
002014
002014 001000
002016
002016 031132
002020
002020 000000
002022
002022 002130
002024
002024 000000
002026
002026 120100
002030
002030 000000
002032
002032 000000
002034
002034 000000
002036
002036 000000
002040
002040 002124
002042
002042 000000
002044
002044 000000
002046

```

;*LAST REVISION 25-MAY-83

.TITLE CZRJKBO RP07 FMTR/SCANNER
.SBTTL PROGRAM HEADER

.ENABL AMA,ABS
      =      2000

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

L$NAME::          ;DIAGNOSTIC NAME
      .ASCII /C/
      .ASCII /Z/
      .ASCII /R/
      .ASCII /J/
      .ASCII /K/
      .BYTE 0
      .BYTE 0
      .BYTE 0

L$REV::          ;REVISION LEVEL
      .ASCII /B/

L$DEPO::         ;0
      .ASCII /O/

L$UNIT::        ;NUMBER OF UNITS
      .WORD T$PTHV

L$TIML::        ;LONGEST TEST TIME
      .WORD 1000

L$HPCP::        ;PTR. TO H.W. QUES.
      .WORD L$HARD

L$SPCP::        ;PTR. TO S.W. QUES.
      .WORD 0

L$HPTP::        ;PTR. TO DEF. H.W. PTABLE
      .WORD L$HW

L$SPTP::        ;PTR. TO S.W. PTABLE
      .WORD 0

L$LADP::        ;DIAG. END ADDRESS
      .WORD L$LAST

L$STA::         ;RESERVED FOR APT STATS
      .WORD 0

L$CO::          ;DIAGNOSTIC TYPE
      .WORD 0

L$DTYP::        ;APT EXPANSION
      .WORD 0

L$APT::         ;PTR. TO DISPATCH TABLE
      .WORD L$DISPATCH

L$DTP::         ;DIAGNOSTIC RUN PRIORITY
      .WORD 0

L$ENVI::        ;FLAGS DESCRIBE HOW IT WAS SETUP
      .WORD 0

L$EXP1::        ;EXPANSION WORD

```

002046	000000			
002050		L\$MREV::	.WORD 0	;SVC REV AND EDIT #
002050	003		.BYTE C\$REVISION	
002051	003		.BYTE C\$EDIT	
002052		L\$EF::	.WORD 0	;DIAG. EVENT FLAGS
002052	000000		.WORD 0	
002054	000000		.WORD 0	
002056		L\$SPC::	.WORD 0	
002056	000000		.WORD 0	
002060		L\$DEVP::	.WORD L\$DVTYP	; POINTER TO DEVICE TYPE LIST
002060	002614		.WORD 0	
002062		L\$REPP::	.WORD 0	;PTR. TO REPORT CODE
002062	000000		.WORD 0	
002064		L\$EXP4::	.WORD 0	
002064	000000		.WORD 0	
002066		L\$EXP5::	.WORD 0	
002066	000000		.WORD 0	
002070		L\$AUT::	.WORD 0	;PTR. TO ADD UNIT CODE
002070	000000		.WORD 0	
002072		L\$DUT::	.WORD 0	;PTR. TO DROP UNIT CODE
002072	000000		.WORD 0	
002074		L\$LUN::	.WORD 0	;LUN FOR EXERCISERS TO FILL
002074	000000		.WORD 0	
002076		L\$DESP::	.WORD L\$DESC	;POINTER TO DIAG. DESCRIPTION
002076	002622		.WORD 0	
002100		L\$LOAD::	EMT E\$LOAD	;GENERATE SPECIAL AUTOLOAD EMT
002100	104035		.WORD 0	
002102		L\$ETP::	.WORD 0	;POINTER TO ERR_TBL
002102	000000		.WORD 0	
002104		L\$ICP::	.WORD L\$INIT	;PTR. TO INIT CODE
002104	026516		.WORD 0	
002106		L\$CCP::	.WORD L\$CLEAN	;PTR. TO CLEAN-UP CODE
002106	027552		.WORD 0	
002110		L\$ACP::	.WORD L\$AUTO	;PTR. TO AUTO CODE
002110	027550		.WORD 0	
002112		L\$PRT::	.WORD L\$PROT	;PTR. TO PROTECT TABLE
002112	026510		.WORD 0	
002114		L\$TEST::	.WORD 0	;TEST NUMBER
002114	000000		.WORD 0	
002116		L\$DLY::	.WORD 0	;DELAY COUNT
002116	000000		.WORD 0	
002120		L\$HIME::	.WORD 0	;PTR. TO HIGH MEM
002120	000000		.WORD 0	

1
2
3
4
5
6
7
8 002122 000001
002124
002124 030074
9

.SBTTL DISPATCH TABLE

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

.WORD 1
L\$DISPATCH:
.WORD T1

```

1      .SBTTL  DEFAULT HARDWARE P-TABLE
2
3
4      ;**
5      ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
6      ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
7      ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
8      ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
9      ;--
10     002126  000004      .WORD  L10000-L$HW/2
11     002130  000004      L$HW::
12     002130  176700      DFPTBL::
13     002132  000254      .WORD  176700      ;RPCS1 BASE REGISTER ADDRESS
14     002134  000240      .WORD  254        ;VECTOR ADDRESS
15     002136  000000      .WORD  240        ;BR LEVEL 5 DEVICE
16     002136  000000      .WORD  0          ;DRIVE NUMBER
17
18
19
20
21
22
23
24
25     002140      L10000:

```

```

1          .SBTTL  SOFTWARE P-TABLE
2
3          ;**
4          ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
5          ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
6          ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
7          ; AT RUN TIME.
8          ;--
9
10         002140  000000          .WORD  L10001-L$SW/2
11         002142
12         002142
13
14
15
16
17
18
19
20         002142          L10001:

```

12
40
50
52
53
54
55
56
57

.SBTTL GLOBAL EQUATES SECTION

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

; BIT DEFINITIONS

100000	BIT15== 100000
040000	BIT14== 40000
020000	BIT13== 20000
010000	BIT12== 10000
004000	BIT11== 4000
002000	BIT10== 2000
001000	BIT09== 1000
000400	BIT08== 400
000200	BIT07== 200
000100	BIT06== 100
000040	BIT05== 40
000020	BIT04== 20
000010	BIT03== 10
000004	BIT02== 4
000002	BIT01== 2
000001	BIT00== 1

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

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

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

; PRIORITY LEVEL DEFINITIONS

000340	PRI07== 340
000300	PRI06== 300
000240	PRI05== 240
000200	PRI04== 200
000140	PRI03== 140
000100	PRI02== 100

000040	PRI01== 40
000000	PRI00== 0
	;
	;OPERATOR FLAG BITS
	;
000004	EVL== 4
000010	LOT== 10
000020	ADR== 20
000040	IDU== 40
000100	ISR== 100
000200	UAM== 200
000400	BOE== 400
001000	PNT== 1000
002000	PRI== 2000
004000	IXE== 4000
010000	IBE== 10000
020000	IER== 20000
040000	LOE== 40000
100000	HOE== 100000


```

1      .SBTTL  RHXX REGISTERS
2
3      ;CONTROL AND STATUS REGISTER 1 (RPCS1)
4
5      000100      INTEN    == 100      ;INTERRUPT ENABLE (BIT #6)
6      000200      RDY      == 200      ;READY (BIT # 7)
7      000400      A16      == 400      ;HIGH ORDER BUS ADDRESS BIT (BIT # 8)
8      001000      A17      == 1000     ;HIGH ORDER BUS ADDRESS BIT (BIT # 9)
9      002000      PSEL     == 2000     ;PORT SELECT (BIT # 10)
10     020000      MCPE     == 20000    ;MASSBUS PARITY ERROR (BIT # 13)
11     040000      TRE      == 40000    ;TRANSFER ERROR (BIT # 14)
12     100000      SC       == 100000   ;SPECIAL CONDITION (BIT # 15)
13
14
15     ;WORD COUNT REGISTER (RPWC)
16     ;EACH BIT IS CALLED BY BIT NUMBER
17
18
19     ;BUS ADDRESS REGISTER (RPBA)
20     ;EACH BIT IS CALLED BY BIT NUMBER
21
22
23     ;CONTROL AND STATUS REGISTER 2 (RPCS2)
24
25     000001      US1      == 1        ;UNIT SELECT (BIT #0)
26     000002      US2      == 2        ;UNIT SELECT (BIT #1)
27     000004      US4      == 4        ;UNIT SELECT (BIT #2)
28     000010      BAI      == 10       ;BUS ADDRESS INCREMENT INHIBIT (BIT #3)
29     000020      PAT      == 20       ;MASSBUS PARITY INHIBIT (BIT #4)
30     000040      CLR      == 40       ;CLEAR (BIT #5)
31     000100      IR       == 100      ;INPUT READY (BIT #6)
32     000200      OR       == 200      ;OUTPUT READY (BIT #7)
33     000400      MPE      == 400      ;MASSBUS PARITY ERROR (BIT #8)
34     001000      MXF      == 1000     ;MISSED TRANSFER ERROR (BIT #9)
35     002000      PGE      == 2000     ;PROGRAM ERROR (BIT #10)
36     004000      NEM      == 4000     ;NON EXISTENT MEMORY (BIT #11)
37     010000      NED      == 10000    ;NON EXISTENT DRIVE (BIT #12)
38     020000      UPE      == 20000    ;UNIBUS PARITY ERROR
39     040000      WCE      == 40000    ;WRITE CHECK ERROR (BIT #14)
40     100000      DLT      == 100000   ;DATA LATE (BIT #15)
41
42
43     ;DATA BUFFER REGISTER (RPDB)
44     ;EACH BIT IS DEFINED BY BIT NUMBER
45
46     .SBTTL  RP07 REGISTERS
47
48     ;CONTROL AND STATUS 1 (#00)
49
50     000001      G0       == 1        ;GO BIT (BIT #0)
51     000002      F1       == 2        ;FUNCTION CODE BIT #1
52     000004      F2       == 4        ;FUNCTION CODE BIT #2
53     000010      F3       == 10       ;FUNCTION CODE BIT #3
54     000020      F4       == 20       ;FUNCTION CODE BIT #4
55     000040      F5       == 40       ;FUNCTION CODE BIT #5
56     004000      DVA      == 4000     ;DEVICE AVAILABLE (BIT #11)
57

```

```

58
59          ;DRIVE STATUS REGISTER (RPDS <#01>)
60
61          000001      OM          == 1          ;OFFSET MODE (BIT #0)
62          000002      EWN         == 2          ;EARLY WARNING (BIT #1)
63          000004      ILEV        == 4          ;INTERLEAVING AVAILABLE (BIT #2)
64          000100      VV          == 100        ;VOLUME VALID (BIT #6)
65          000200      DRY         == 200        ;DATA READY (BIT #7)
66          000400      DPR         == 400        ;DRIVE PRESENT (BIT #8)
67          001000      PGM         == 1000       ;PROGRAMABLE (BIT #9)
68          002000      LBT         == 2000       ;LAST BLOCK TRANSFERRED (BIT #10)
69          004000      WRL         == 4000       ;WRITE LOCKED (BIT #11)
70          010000      MOL         == 10000      ;MEDIUM ON LINE (BIT #12)
71          020000      PIP         == 20000     ;POSITIONER IN PROGRESS (BIT #13)
72          040000      ERR         == 40000     ;COMPOSITE ERROR (BIT #14)
73          100000      ATA         == 100000    ;ATTENTION ACTIVE (BIT #15)
74
75
76          ;ERROR REGISTER #1 (RPER1 <#02>)
77
78          000001      ILF         == 1          ;ILLEGAL FUNCTION (BIT #0)
79          000002      ILR         == 2          ;ILLEGAL REGISTER (BIT #1)
80          000004      RMR         == 4          ;REGISTER MODIFICATION REFUSED (BIT #2)
81          000010      PAR         == 10         ;PARITY ERROR (BIT #3)
82          000020      FER         == 20         ;FORMAT ERROR (BIT #4)
83          000040      WCF         == 40         ;WRITE CLOCK FAIL (BIT #5)
84          000100      ECH         == 100        ;ECC HARD ERROR (BIT #6)
85          000200      HCE         == 200        ;HEADER COMPARE ERROR (BIT #7)
86          000400      HCRC        == 400        ;HEADER CRC ERROR (BIT #8)
87          001000      AOE         == 1000       ;ADDRESS OVERFLOW ERROR (BIT #9)
88          002000      IAE         == 2000       ;INVALID ADDRESS ERROR (BIT #10)
89          004000      WLE         == 4000       ;WRITE LOCK ERROR (BIT #11)
90          010000      DTE         == 10000     ;DRIVE TIMING ERROR (BIT #12)
91          020000      OPI         == 20000     ;OPERATION INCOMPLETE (BIT #13)
92          040000      UNS         == 40000     ;DRIVE UNSAFE (BIT #14)
93          100000      DCK         == 100000    ;DATA CHECK ERROR (BIT #15)
94
95
96          ;DIAGNOSTIC MAINTAINABILTY REGISTER (RPMR1 <#03>)
97
98          100000      DMD         == 100000     ;DIAGNOSTIC MODE (BIT #15)
99
100
101          ;ATTENTION SUMMARY PSEUDO REGISTER (RPAS <#04>)
102
103          000001      AT0         == 1          ;DEVICE 0 (BIT #0)
104          000002      AT1         == 2          ;DEVICE 1 (BIT #1)
105          000004      AT2         == 4          ;DEVICE 2 (BIT #2)
106          000010      AT3         == 10         ;DEVICE 3 (BIT #3)
107          000020      AT4         == 20         ;DEVICE 4 (BIT #4)
108          000040      AT5         == 40         ;DEVICE 5 (BIT #5)
109          000100      AT6         == 100        ;DEVICE 6 (BIT #6)
110          000200      AT7         == 200        ;DEVICE 7 (BIT #7)
111
112
113          ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA <#05>)
114          ;EACH BIT IS CALLED BY BIT NUMBER

```

```

115
116
117           ;DRIVE TYPE REGISTER (RPDT <#06>)
118
119           000001      DRT0    == 1           ;DRIVE TYPE NUMBER (BIT #0)
120           000002      DRT1    == 2           ;DRIVE TYPE NUMBER (BIT #1)
121           000004      DRT2    == 4           ;DRIVE TYPE NUMBER (BIT #2)
122           000010      DRT3    == 10          ;DRIVE TYPE NUMBER (BIT #3)
123           000020      DRT4    == 20          ;DRIVE TYPE NUMBER (BIT #4)
124           000040      DRT5    == 40          ;DRIVE TYPE NUMBER (BIT #5)
125           000100      DRT6    == 100         ;DRIVE TYPE NUMBER (BIT #6)
126           000200      DRT7    == 200         ;DRIVE TYPE NUMBER (BIT #7)
127           000400      DRT8    == 400         ;DRIVE TYPE NUMBER (BIT #8)
128           004000      DRQ     == 4000        ;DRIVE REQUEST REQUIRED (BIT #11)
129           020000      MOH     == 20000       ;MOVING HEAD TYPE DRIVE (BIT #13)
130           040000      TAP     == 40000       ;TAPE DRIVE (BIT #14)
131           100000      NBA     == 100000      ;NOT BLOCK ADDRESSED (BIT #15)
132
133
134           ;LOOK AHEAD REGISTER (RPLA <#07>)
135
136           000100      SC1     == 100          ;SECTOR COUNT FIELD 1 (BIT #6)
137           000200      SC2     == 200          ;SECTOR COUNT FIELD 2 (BIT #7)
138           000400      SC4     == 400          ;SECTOR COUNT FIELD 4 (BIT #8)
139           001000      SC8     == 1000         ;SECTOR COUNT FIELD 8 (BIT #9)
140           002000      SC16    == 2000        ;SECTOR COUNT FIELD 16 (BIT #10)
141           004000      SC32    == 4000        ;SECTOR COUNT FIELD 32 (BIT #11)
142           010000      SC64    == 10000       ;SECTOR COUNT FIELD 64 (BIT #12)
143
144
145           ;RP07 SERIAL NUMBER REGISTER (RPSN <#10>)
146           ;EACH BIT IS CALLED BY BIT NUMBER
147
148
149           ;RP07 OFFSET REGISTER (RPOF <#11>)
150
151           000200      OFFDIR   == 200          ;OFFSET DIRECTION (BIT #7)
152           002000      HCI      == 2000        ;HEADER COMPARE CODE INHIBIT (BIT #10)
153           004000      ECI      == 4000        ;ERROR CORRECTION CODE INHIBIT (BIT #11)
154           010000      FMT16    == 10000       ;16 BIT FORMAT (BIT #12)
155           040000      MTD      == 40000       ;MOVE TRACK DESCRIPTOR (BIT #14)
156           100000      CMOD     == 100000      ;COMMAND MODIFIER (BIT #15)
157
158
159           ;RP07 DESIRED CYLINDER ADDRESS (RPDC <#12>)
160           ;EACH BIT IS CALLED BY BIT NUMBER
161
162
163           ;RP07 CURRENT CYLINDER ADDRESS (RPCC <#13>)
164           ;EACH BIT IS CALLED BY BIT NUMBER
165
166
167           ;RP07 ERROR REGISTER 3 (RPER3 <#15>)
168
169           000002      SCF      == 2           ;SYNC CLOCK FAILURE (BIT #1)
170           000004      SBE      == 4           ;SYNC BYTE ERROR (BIT #2)
171           000010      DPE      == 10          ;DATA PARITY ERROR (BIT #3)

```

```

172      000020      SDF      == 20      ;SERDES DATA FAILURE (BIT #4)
173      000040      DCU       == 40      ;DC UNSAFE (BIT #5)
174      000100      IXU       == 100     ;INDEX UNSAFE (BIT #6)
175      000200      DVC       == 200     ;DEVICE CHECK (BIT #7)
176      000400      PHF       == 400     ;8080 PROCESSOR HANDSHAKE FAILURE (BIT #8)
177      001000      LCE       == 1000    ;LOSS OF CYLINDER ERROR (BIT #9)
178      002000      LBC       == 2000    ;LOSS OF BIT CLOCK (BIT #10)
179      020000      DSE       == 20000   ;DEFECT SKIP ERROR (BIT #13)
180      040000      SKI       == 40000   ;SEEK INCOMPLETE (BIT #14)
181      100000      BSE       == 100000  ;BAD SECTOR ERROR (BIT #15)
182
183
184      ;RP07 ERROR REGISTER #2 (RPER2 <#14>)
185
186      000400      WRU       == 400     ;WRITE READY UNSAFE (BIT #8)
187      001000      WOR       == 1000    ;WRITE OVERRUN (BIT #9)
188      002000      RWU1      == 2000    ;READ/WRITE UNSAFE #1 (BIT #10)
189      004000      RWU2      == 4000    ;READ/WRITE UNSAFE #2 (BIT #11)
190      010000      RWU3      == 10000   ;READ/WRITE UNSAFE #3 (BIT #12)
191      020000      CPU       == 20000   ;CPU UNSAFE (BIT #13)
192      040000      CPE       == 40000   ;CROM PARITY ERROR (BIT #14)
193      100000      PGE       == 100000  ;PROGRAMING ERROR
194
195
196      ;ECC POSITION REGISTER (RPEC1 <#16>)
197      ;EACH BIT IS DEFINED BY BIT NUMBER
198
199
200      ;ECC PATTERN REGISTER (RPEC2 <#17>)
201      ;EACH BIT IS DEFINED BY BIT NUMBER
202
203      .SBTTL  RP07 DRIVER COMMANDS
204
205      000001      NOOP      == 1      ;NO OPERATION
206      000005      SEEK      == 5      ;SEEK
207      000007      RECAL     == 7      ;RECALIBRATE
208      000011      DRVCLR    == 11     ;DRIVE CLEAR
209      000013      RELSE     == 13     ;RELEASE
210      000021      READIN    == 21     ;READ IN PRESET
211      000031      SEARCH    == 31     ;SEARCH
212      000035      DIAG      == 35     ;DIAGNOSTIC MODE
213      000051      WCKD      == 51     ;WRITE CHECK DATA
214      000053      WCKHD     == 53     ;WRITE CHECK HEADER & DATA
215      000061      WRTDAT    == 61     ;WRITE DATA
216      000063      FMTRK     == 63     ;FORMAT TRACK
217      000065      WRTTD     == 65     ;WRITE TRACK DESCRIPTOR
218      000071      RDDAT     == 71     ;READ DATA
219      000073      RDHD      == 73     ;READ HEADER & DATA
220      000075      RDTD      == 75     ;READ TRACK DESCRIPTOR
221
222      ;SOME TRACK FORMAT EQUATES FOR 16 BIT MODE (IN WORDS)
223
224
225
226
227
228      000100      G1        == 64.    ;SIZE OF GAP 1
229      000010      TD        == 8.     ;SIZE OF TD
230      000045      G2        == 37.    ;SIZE OF GAP 2
231      000010      HDR0      == 8.     ;SIZE OF HEADER 0
232      000010      HDR       == 8.     ;SIZE OF HEADER
  
```

233	000040	G3	== 32.	;SIZE OF GAP 3
234	000402	DATA	== 258.	;SIZE OF DATA FIELD
235	000045	G4	== 37.	;SIZE OF GAP 4
236				
237	000154	DS	== 108.	;SIZE OF DEFECT SKIP
238	000022	MSEG	== 18.	;SIZE OF MINIMUM DATA SEGMENT
239	140000	NULL	== 140000	;NULL DATA FOR TRACK DESCRIPTOR
240				

```

1      .SBTTL  GLOBAL DATA SECTION
2
3
4      ;**
5      ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
6      ; IN MORE THAN ONE TEST.
7      ;--
8      BGNPAT  ==.
9      002142 000000      .WORD 000000      ;PATTERN #1 (ALL 0'S)
10     002144 000000      .WORD 000000
11     002146 177777      .WORD 177777      ;PATTERN #2 (ALL 1'S)
12     002150 177777      .WORD 177777
13     002152 030221      .WORD 030221      ;PATTERN #3 (WORST CASE)
14     002154 030221      .WORD 030221
15     002156 002156
16     ENDPAT  ==.
17
18     002156 000000      ANYWHR:: .WORD 0      ;IF = 1, ALLOW ACCESS TO ANYWHERE ON MEDIA
19                                     ;IF = 0, ALLOW ACCESS TO FE CYLINDER 1 ONLY
20
21     002160 000000      OPTION:: .WORD 0      ;OPTIONS; 0=FORMAT, 1=VERIFY, 2=SCAN, 3=MODIFY,
22                                     ; 4=LIST, 5=WRITE FE-2 AND 6=HELP
23     002162 000000      FORMT:: .WORD 0      ;FORMAT OPTION; ENABLED= 1, DISABLED= 0
24     002164 000000      VRIFY:: .WORD 0      ;VERIFY OPTION; ENABLED= 1, DISABLED= 0
25     002166 000000      SCANR:: .WORD 0      ;SCAN OPTION; ENABLED= 1, DISABLED= 0
26     002170 000000      MODTD:: .WORD 0      ;MODIFY OPTION; ENABLED= 1, DISABLED= 0
27     002172 000000      LIST:: .WORD 0      ;LIST OPTION; ENABLED= 1, DISABLED= 0
28     002174 000000      WRTFE2:: .WORD 0      ;WRITE 2ND FE CYL OPTION; ENABLED= 1, DISABLED= 0
29
30     002176 000000      NOWRCK:: .WORD 0      ;USED TO INHIBIT WRITE CHECK COMMAND
31     002200 000000      ENWTTD:: .WORD 0      ;USED TO ENABLE WRITE TD OPTION AFTER PACK SCAN
32     002202 000000      MODBY:: .WORD 0      ;MODIFY TD BY; 0=WORDS, 1=INDEX, 2=CHANGE
33     002204 000000      LISHDR:: .WORD 0      ;USED TO LIST HEADER INFO IN DEFECT SECTORS
34     002206 000000      DRVPAR:: .WORD 0      ;WHEN EQ TO 1 CHANGE DRIVE PARAMETERS
35
36     002210 000000      MINCYL:: .WORD 0      ;USED TO DETERMINE MIN CYLINDER OF OPERATION
37     002212 001166      MAXCYL:: .WORD 630.    ;USED TO DETERMINE MAX CYLINDER OF OPERATION
38     002214 001166      TMPCYL:: .WORD 630.    ;TEMPORARY STORAGE FOR MAXIMUM CYLINDER VALUE
39     002216 000000      MINTRK:: .WORD 0      ;USED TO DETERMINE MIN TRACK OF OPERATION
40     002220 000037      MAXTRK:: .WORD 31.    ;USED TO DETERMINE MAX TRACK OF OPERATION
41
42     002222 001165      LSCYL:: .WORD 629.    ;LAST USER CYLINDER ADDRESS
43     002224 000037      LSTRK:: .WORD 31.    ;LAST TRACK ADDRESS
44     002226 000005      ERRMAX:: .WORD 5      ;USED TO CONTROL MAX ERRORS
45     002230 000310      ENDPTR:: .WORD 50.*4  ;USED TO CREATE BUFFER BOUNDARIES
46     002232 000000      ENDTAB:: .WORD 0      ;USED TO MARK BUFFER BOUNDARIES
47     002234 000000      BUSADR:: .WORD 0      ;CONTAINS BUS ADDR FOR DRIVER MODULE
48     002236 000000      OFFSET:: .WORD 0      ;CONTENTS OF OFFSET REGISTER FOR DRIVER MODULE
49     002240 032464      LASLOC:: .WORD DEFBUF ;USED TO LINK BUFFER TO BUFFER TRANSFERS
50
51     002242 000      FIRPAS:: .BYTE 0      ;USED TO CONTROL SCANNER ITERATIONS
52     002243 000      SCANIT:: .BYTE 0      ;# OF SCAN ITERATIONS DURING A SCAN OPERATION
53     002244 000      INTLEV:: .BYTE 0      ;INTERLEAVE MODE INDICATOR
54     002245 000      SAMSEC:: .BYTE 0      ;# DEFECTS IN CURRENT SECTOR
55     002246 000      NEWCNT:: .BYTE 0      ;USED TO TRACK NEWLY FOUND DEFECTS
56     002247 000      CONFLG:: .BYTE 0      ;SET IF PROGRAM IS DOING CONTINUE COMMAND
57                                     .EVEN
58
59     002250 000000      TEMPA:: .WORD 0      ;USED FOR SOFTWARE CALCULATIONS
60     002252 000000      SUPRSS:: .WORD 0      ;CONTROL USED TO SUPPRESS ERROR MESSAGES
    
```

```

62 002254 000000 DEFBCNT:: .WORD 0 ;USED TO TRACK THE HDA DEFECT COUNT
63 002256 000000 TBLPTR:: .WORD 0 ;USED TO FEED AN ADDRESS TO 'GETNEX'
64 002260 000000 ERRFNC:: .WORD 0 ;USED TO STORE THE FUNCTION AT TIME OF FAILURE
65 002262 000000 SOFSW:: .WORD 0 ;USED FOR SOFTWARE SCRATCH
66 002264 000000 ITCNT:: .WORD 0 ;CONTAINS # OF ITERATIONS TO MAKE ON A COMMAND
67 002266 000000 ERTTL:: .WORD 0 ;CONTAINS THE TOTAL # OF REPORTED ERRORS DETECTED
68 002270 000000 ERRCTL:: .WORD 0 ;CONTAINS THE TOTAL # OF ERRORS ON THE DEVICE
69 002272 000000 ERRMSK:: .WORD 0 ;ERROR MASK
70 002274 000000 RTYCNT:: .WORD 0 ;USED TO MAINTAIN THE LOCAL RETRY COUNTER
71 002276 000000 EXPCTD:: .WORD 0 ;USED TO IDENTIFY EXPECTED DATA
72 002300 000000 RECVD:: .WORD 0 ;USED TO IDENTIFY RECEIVED DATA
73 002302 000000 DEF1:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH ONE DEFECT
74 002304 000000 DEF2:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH TWO DEFECTS
75 002306 000000 DEF3:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH THREE DEFECTS
76 002310 000000 DEF4:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH FOUR DEFECTS
77 002312 000000 DESCYL:: .WORD 0 ;DESIRED CYLINDER ADDRESS
78 002314 000000 DESTRK:: .WORD 0 ;DESIRED TRACK ADDRESS (IN HI BYTE)
79 ;DESIRED SECTOR ADDRESS (IN LO BYTE)
80 002316 000000 SAVCYL:: .WORD 0 ;SAVED CYLINDER ADDRESS
81 002320 000000 SAVTRK:: .WORD 0 ;SAVED TRACK ADDRESS (IN HI BYTE)
82 ;SAVED SECTOR ADDRESS (IN LO BYTE)
83 002322 061 SEC50:: .BYTE 49. ;50 SECTORS (0-49.)
84 002323 031 000 PLTRK:: .BYTE 25..0 ;INTERLEAVED FILE FOR 16-BIT FORMAT
85 ;.EVEN
86 002326 002142 PATTRN:: .WORD BGNPAT ;POINTER TO BEGINNING OF DATA PATTERNS
87 002330 000006 TDBCNT:: .WORD 6 ;TD BYTE COUNT
88 002332 000110 TDVALU:: .WORD G1+TD ;GAP 1 + TD CONSTANT
89 002334 000045 .WORD G2 ;GAP 2 CONSTANT
90
91 ;*****
92 ; THE FOLLOWING TABLE REPRESENTS THE VARIOUS PARTITIONS WITHIN A SECTOR FOR AN
93 ; RP07. (IN 16 BIT MODE) THE TABLE STRUCTURE MUST NOT BE ALTERED IN ANYWAY.
94 ;
95 ; >>>DO NOT DISTURB THE FOLLOWING TABLE UNDER ANY CIRCUMSTANCES<<<
96 ;
97 002336 000010 PARTBL:: .WORD HDR ;SECTOR HEADER,
98 002340 000040 .WORD G3 ;GAP 3,
99 002342 000022 .WORD MSEG ;LEADING MIN. DATA SEGMENT,
100 002344 000336 .WORD 222. ;DATA FIELD SEGMENT,
101 002346 000022 .WORD MSEG ;TRAILING MIN. DATA SEGMENT,
102 002350 000045 .WORD G4 ;GAP 4.
103 ;*****
104 ;*****
105 ; THE TWO TD LIMIT WORDS, 'TDLMTS' AND 'TDLMTS+2' ARE REFERENCED FROM
106 ; INDEX TO THE CENTER OF THE FIRST DEFECT SKIP (DS1). WHILE THE TD LIMIT
107 ; WORD 'TDLMTS+4' IS REFERENCED FROM THE CENTER OF THE FIRST DEFECT SKIP (DS1)
108 ; TO THE CENTER OF THE SECOND DEFECT SKIP (DS2). THESE NUMBERS WILL BE USED
109 ; WHEN THE PROGRAM IS TRYING TO DETERMINE IF A TD AND/OR HDR 0 SHOULD
110 ; BE MOVED.
111 ;
112 ;
113 002352 000166 TDLMTS:: .WORD G1+<DS/2> ;TD MOVED LIMIT
114 002354 000243 .WORD G1+TD+G2+<DS/2> ;HDR 0 MOVED LIMIT
115 002356 000231 .WORD <DS/2>+TD+G2+<DS/2> ;TD & HDR 0 MOVED LIMIT
116 ;*****
117 ;*****
118 002360 000000 TEXT:: .WORD 0 ;POINTS TO ADDRESS OF TEXT TO BE TYPED

```

119	002362	000000	CONTLT:: .WORD	0	; ADDRESS OF CONTROL 'T' SERVICE BUFFER
120	002364	000454	TKWCNT:: .WORD	6*50.	; WORDS FOR TRACK FORMAT (6 WORDS X 50. SECTORS)
121	002366	000000	TEMP1:: .WORD	0	; USED TO IDENTIFY SCRATCHES
122	002370	000000	TEMP2:: .WORD	0	; USED TO MEASURE SCRATCH LENGTH
123	002372	000000	HICYL:: .WORD	0	; USED TO MAP THE HI ADDRESS OF A SCRATCH
124	002374	000000	LOCYL:: .WORD	0	; USED TO MAP THE LO ADDRESS OF A SCRATCH
125	002376	000000	NEGWRD:: .WORD	0	; NEGATED WORD COUNT FOR DRIVER
126	002400	000000	RELWRD:: .WORD	0	; CONTAINS # OF WORDS TO DEFECT, RELATIVE TO INDEX
127	002402	000000	SECADD:: .WORD	0	; SECTOR ADDRESS OF NEW TD DEFECT
128	002404	000000	FUNCTN:: .WORD	0	; FUNCTION COMMAND FOR RP07 DRIVER MODULE
129	002406	000000	DEFSEC:: .WORD	0	; CONTAINS # OF DEFECTS IN A SECTOR
130	002410	000000	DEFTRK:: .WORD	0	; CONTAINS # OF DEFECTS ON A TRACK
131	002412	000000	MINSEG:: .WORD	0	; MINIMUM VALUE FOR A DATA SEGMENT
132	002414	000000	MAXSEG:: .WORD	0	; MAXIMUM VALUE FOR A DATA SEGMENT
133	002416	000000	MORETD:: .WORD	0	; IF EQ 1, THEN MORE TD'S TO MODIFY
134					
135	002420	000000	UNIT:: .WORD	0	; USED TO SELECT A UNIT NUMBER
136	002422	176700	RPADR:: .WORD	176700	; CONTAINS RPCS1 BASE ADDRESS
137	002424	000254	RPVEC:: .WORD	254,5*32.	; CONTAINS VECTOR ADDRESS & BR LEVEL
138	002430	000050	RHEXT:: .WORD	50	; CONTAINS RH70 OFFSET TO RPBAE
139	002432	000000	RHTYPE:: .WORD	0	; CONTAINS RHXX TYPE; RH11= 0, RH70= 1
140	002434	000000	DRVNO:: .WORD	0	; DRIVE NUMBER
141	002436	000000	DRVSN:: .WORD	0	; STORAGE FOR EACH S/N DIGIT
142					
143	002440	176700	RPCS1:: .WORD	176700	; BASE ADDRESS USED FOR THE DRIVE
144	002442	176702	RPWC:: .WORD	176702	; WORD COUNT REGISTER
145	002444	176704	RPBA:: .WORD	176704	; BYTE ADDRESS REGISTER
146	002446	176706	RPDA:: .WORD	176706	; DESIRED SECTOR/TRACK ADDRESS
147	002450	176710	RPCS2:: .WORD	176710	; RP07 STATUS REGISTER
148	002452	176712	RPDS:: .WORD	176712	; RP07 DRIVE STATUS
149	002454	176714	RPER1:: .WORD	176714	; RP07 ERROR REGISTER #1
150	002456	176716	RPAS:: .WORD	176716	; RP07 ATTENTION SUMMARY PSEUDO REGISTER
151	002460	176720	RPLA:: .WORD	176720	; RP07 LOOK AHEAD REGISTER
152	002462	176722	RPDB:: .WORD	176722	; RP07 DATA BUFFER
153	002464	176724	RPMR1:: .WORD	176724	; RP07 MAINTENANCE REGISTER #1
154	002466	176726	RPDT:: .WORD	176726	; DRIVE TYPE REGISTER
155	002470	176730	RPSN:: .WORD	176730	; RP07 SERIAL NUMBER
156	002472	176732	RPOF:: .WORD	176732	; RP07 OFFSET REGISTER
157	002474	176734	RPDC:: .WORD	176734	; RP07 DESIRED CYLINDER
158	002476	176736	RPCC:: .WORD	176736	; RP07 CURRENT CYLINDER
159	002500	176740	RPER2:: .WORD	176740	; RP07 ERROR REGISTER #2
160	002502	176742	RPER3:: .WORD	176742	; RP07 ERROR REGISTER #3
161	002504	176744	RPEC1:: .WORD	176744	; RP07 ERROR-POSITION
162	002506	176746	RPEC2:: .WORD	176746	; RP07 ERROR PATTERN
163	002510	176750	RPBAE:: .WORD	176750	; RH70 REGISTER
164	002512	176752	RPCS3:: .WORD	176752	; RH70 REGISTER
165					
166			; STORAGE FOR DEVICE REGISTERS		
167					
168	002514		REG:: .BLKW	22.	; BUFFER TO SAVE REGISTERS AFTER AN ERROR
169					
170	002570		DELTA:: .BLKW	4	; BUFFER USED TO STORE DELTA ADJUSTMENTS
171	002600		CMDQUE:: .BLKW	6	; USED FOR A COMMAND SEQUENCE QUEUE
172					


```

1      .SBTTL  GLOBAL TEXT SECTION
2
3
4      ; **
5      ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
6      ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
7      ; MORE THAN ONE TEST.
8      ; --
9
10     ; NAMES OF DEVICES SUPPORTED BY PROGRAM
11     ;
12     L$DVTYP::
13     .ASCIZ  /RP07/
14     .EVEN
15
16
17     ; TEST DESCRIPTION
18     ;
19     L$DESC::
20     .ASCIZ  /RP07 FORMAT-VERIFY-SCANNER/
21     .EVEN
22
23
24     ; FORMAT STATEMENTS USED IN PRINT CALLS
25     ;
26
27     FRMT00:: .ASCIZ  /#N#ADRIIVE  RPCS1  RPWC  RPBA  RPDA  RPCS2  RPDS/
28     FRMT01:: .ASCIZ  /#N#06#A #06#A #06#A #06#A #06#A #06#A #06/
29     FRMT02:: .ASCIZ  /#N#ARPER1  RPAS  RPLA  RPD8  RPMR1  RPDT  RPSN/
30     FRMT03:: .ASCIZ  /#N#06#A #06#A #06#A #06#A #06#A #06#A #06/
31     FRMT04:: .ASCIZ  /#N#ARPOF  RPDC  RPCC  RPER2  RPER3  RPEC1  RPEC2/
32     FRMT05:: .ASCIZ  /#N#06#A #06#A #06#A #06#A #06#A #06#A #06#N/
33     FRMT06:: .ASCIZ  /#ARPBAE  RPCS3/
34     FRMT07:: .ASCIZ  /#N#06#A #06#N/
35
36     FRMT10:: .ASCIZ  /#N#ADRIIVE #01/
37     FRMT11:: .ASCIZ  /#N#ADONE.  RETRIES MADE= #D5#A.,  ERRORS DETECTED= #D5#A.#N/
38     FRMT12:: .ASCIZ  /#ATRACKS WITH #D2#A DEFECTS= #D5#A.#N/
39     FRMT13:: .ASCIZ  /#N#ATOTAL DEFECTS FOUND= #D5#A./
40     FRMT14:: .ASCIZ  /#D5#A. /
41     FRMT15:: .ASCIZ  /#06#A /
42     FRMT16:: .ASCIZ  /#N#ADRIIVE  WORD#1  WORD#2  WORD#3  WORD#4  WORD#5  WORD#6/
43     FRMT17:: .ASCIZ  /#N#06#A #06#A #06#A #06#A #06#A #06#A #06/
44
45     FRMT20:: .ASCIZ  /#N#ACYL:#D3#A.  TRK:#D2#A. /
46     FRMT22:: .ASCIZ  /#APRSNT FUNCT: #T/
47     FRMT23:: .ASCIZ  /#T#A #06#A #06/
48     FRMT24:: .ASCIZ  /#N#ATYPE <CR> TO INPUT (0=140000) DATA;/
49
50     FRMT30:: .ASCIZ  /#A, NON-INTERLEAVED#N/
51     FRMT31:: .ASCIZ  /#A, INTERLEAVED#N/
52     FRMT32:: .ASCIZ  /#N#ANEW DEFECT(S) DURING SCAN ITERATION #D3#A., TD(S) FOLLOW:/
53     FRMT33:: .ASCIZ  /#N#AEXPCTD:#06#A  RECVED:#06/
54     FRMT35:: .ASCIZ  /#N#AWORD#1 ( cyl)  WORD#2 (trk)(sec)  WORD#3  WORD#4  WORD#5  WORD#6#N/
55     FRMT36:: .ASCIZ  /#06#A (#D3#A.) #06#A (#D2#A.)(#03#A) /
56     FRMT37:: .ASCIZ  /#A #06#A (#D2#A.) /
    
```

```

71
72 004634      045      116      045  FRMT50::.ASCII  /%N%AAVAILABLE OPTIONS FOLLOW;/
73 004671      045      116      045      .ASCII  /%N%A 0 =Format/
74 004707      045      116      045      .ASCII  /%N%A 1 =Verify/
75 004725      045      116      045      .ASCIZ  /%N%A 2 =Scan/
76 004742      045      116      045  FRMT51::.ASCII  /%N%A 3 =List/
77 004756      045      116      045      .ASCII  /%N%A 4 =Modify/
78 004774      045      116      045      .ASCIZ  /%N%A 5 =Write FE-2%N/
79
80      .SBTTL  WARNING MESSAGES
81
82 005021      045      123      061  FRMT60:: .ASCIZ  /%S11%AINput data rejected%N/
83 005055      045      116      045  FRMT61:: .ASCIZ  /%N%T%ACYL %D3%A., TRK %D2%A. has 4 defects in track descriptor!%N/
84 005157      045      116      045  FRMT62:: .ASCIZ  /%N%T%AThere are %D5%A. track(s) with 4 defects!%N/
85 005241      045      116      045  FRMT63:: .ASCIZ  /%N%T%ABad sector file buffer is full, no more entries allowed!%N/
86 005342      045      116      045  FRMT64:: .ASCIZ  /%N%T%ABad sector file data is corrupt, it will be initialized!%N/
87 005443      045      123      061  FRMT65:: .ASCIZ  /%S11%AAanother defect was found %D5%A. words from Index%N/
88 005534      045      123      061  FRMT66:: .ASCIZ  /%S11%ABad sector file will be updated%N/
89
90      .SBTTL  GLOBAL ASCII MESSAGE SECTION
91
92 005604      127      122      111  MSWRTD:: .ASCIZ  /WRITE TD/
93 005615      122      105      101  MSRDTD:: .ASCIZ  /READ TD/
94 005625      127      122      124  WRDAT:: .ASCIZ  /WRT DATA/
95 005636      127      122      124  WCKDAT:: .ASCIZ  /WRT CHK DATA/
96 005653      127      122      124  WCKHDR:: .ASCIZ  /WRT CHK HEADER/
97 005672      122      105      103  RECALI:: .ASCIZ  /RECAL/
98 005700      106      117      122  FORMAT:: .ASCIZ  /FORMAT TRK/
99 005713      122      105      101  RHDATA:: .ASCIZ  /READ HEADER/
100 005727      124      105      123  MOLINE:: .ASCIZ  /TEST MOL = 1/
101 005743      124      105      123  MLOCK:: .ASCIZ  /TEST WRL = 0/
102 005757      124      105      123  DRVRDY:: .ASCIZ  /TEST RPDS: BIT7 = 1/
103 006002      015      012      105  EXPTD:: .ASCIZ  <CR><LF>/EXPCTD: /
104 006015      015      012      122  RCVED:: .ASCIZ  <CR><LF>/RCVED: /
105 006030      015      012      103  CURENT:: .ASCIZ  <CR><LF>/CURRENT TD./
106 006046      015      012      124  TODEF:: .ASCIZ  <CR><LF>/TD(S) WITH DEFECTS FOLLOW:/
107 006103      015      012      124  ANDHDR:: .ASCIZ  <CR><LF>/TD(S) WITH DEFECTS (INCLUDING HEADER INFO) FOLLOW:/
108 006170      015      012      103  CHANGE:: .ASCIZ  <CR><LF>/CHANGED TD./
109 006206      127      101      122  WARN:: .ASCIZ  /WARNING ** /<BELL><BELL>
110
111 006224      103      110      101  MMSG18:: .ASCIZ  /CHANGE DRIVE PARAMETERS/
112 006254      105      116      124  MMSG7:: .ASCIZ  /ENTER OPTION (6=HELP)/
113 006302      115      111      116  MMSG13:: .ASCIZ  /MIN TRK/
114 006312      115      101      130  MMSG14:: .ASCIZ  /MAX TRK/
115 006322      115      111      116  MMSG15:: .ASCIZ  /MIN CYL/
116 006332      115      101      130  MMSG16:: .ASCIZ  /MAX CYL/
118 006342      104      117      040  WRITMG: .ASCIZ  /DO YOU WANT TO WRITE ANYWHERE ON MEDIA/
119 006411      007      011      041  WRSAFM: .ASCII  <BELL>/ ! CUSTOMER DATA WILL BE OVERWRITTEN !/<CR><LF>
120 006462      007      011      055      .ASCII  <BELL>/ -----/<CR><LF>
121 006533      103      117      116      .ASCIZ  /CONTINUE/
123 006544      111      116      110  MMSG17:: .ASCIZ  /INHIBIT WRITE CHECK/
124 006570      115      117      104  MMSG22:: .ASCIZ  /MODIFY BY (0 =Words, 1 =Index, 2 =Change)/
125 006642      104      117      040  MMSG19:: .ASCIZ  /DO YOU WANT TO RE-WRITE TD(S) WITH NEW DEFECTS/
126 006721      114      111      123  MMSG23:: .ASCIZ  /LIST HEADER INFO IN DEFECT SECTOR(S)/
127
128 006766      124      104      040  EM1:: .ASCIZ  /TD ADDRESS INCORRECT/
129 007013      124      104      040  EM2:: .ASCIZ  /TD FORMAT INCORRECT/
    
```

130	007037	103	117	115	EM3::	.ASCIZ	/COMPOSITE ERROR SET/
131	007063	104	122	111	EM4::	.ASCIZ	/DRIVE HUNG, DRY NOT SET IN TIME/
132	007123	104	122	111	EM5::	.ASCIZ	/DRIVE WRITE LOCKED/
133	007146	104	122	111	EM6::	.ASCIZ	/DRIVE OFFLINE/
134	007164	125	116	105	EM7::	.ASCIZ	/UNEXPECTED ATTN OCCURRED/
135							
136	007215	127	122	111	EM10::	.ASCIZ	/WRITE CHECK ERROR SET/
137	007243	106	117	122	EM11::	.ASCIZ	/FORMAT VERIFY ERROR/
138	007267	104	122	131	EM12::	.ASCIZ	/DRY DIDN'T RESET WHEN EXPECTED/
139	007326	124	122	101	EM13::	.ASCIZ	/TRANSFER ERROR SET/
140	007351	123	120	105	EM14::	.ASCIZ	/SPECIAL CONDITION SET/
141	007377	122	101	116	EM15::	.ASCIZ	/RANDOM HEADER ERRORS -CAN'T RESOLVE DEFECTS-/
142	007454	103	117	125	EM16::	.ASCIZ	/COULD NOT SUPPLY READABLE TD THRU RELOCATION/
143	007531	125	116	122	EM17::	.ASCIZ	/UNRECOVERABLE ERROR DURING PACK SCAN/
144							
145	007576	122	101	116	EM20::	.ASCIZ	/RANDOM WRT CHK ERRORS -CAN'T RESOLVE DEFECTS-/
146	007654	116	117	116	EM21::	.ASCIZ	/NON-EXISTENT DRIVE/
147	007677	104	122	111	EM22::	.ASCIZ	/DRIVE NOT AN RP07/
148						.EVEN	
152							
163							

```

1      .SBTTL  GLOBAL ERROR REPORT SECTION
2
3
4      ;**
5      ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX CALLS
6      ; THAT ARE USED IN MORE THAN ONE TEST.  IT ALSO INCLUDES THE ASCII MESSAGES
7      ; THAT ARE USED BY THE PRINTB AND PRINTX CALLS..
8      ;--
9
10     007722      004737  011046      ERRO::      JSR      PC, TYPLOC      ;REPORT THE ADDRESS OF THE FAILURE
11     007726      004737  011132      JSR      PC, DMPREG      ;DUMP THE ERROR REGISTERS
12
13     007732      012746  002656      MOV      @CRLF, -(SP)      ;CR-LF
14     007736      012746  000001      MOV      @1, -(SP)
15     007742      010600      MOV      SP, R0
16     007744      104414      TRAP    C#PNTB
17     007746      062706  000004      ADD     @4, SP
18
19     007752      104423      L10002: TRAP    C#MSG
20
21     007754      004737  011046      ERR1::     JSR      PC, TYPLOC      ;REPORT THE ADDRESS OF THE FAILURE
22     007760      013746  031432      MOV      TDCPY2, -(SP)      ;PRINT 'EXPCTD: XXXXXX XXXXXX'
23     007764      013746  031430      MOV      TDCPY1, -(SP)
24     007770      012746  006002      MOV      @EXPTD, -(SP)
25     007774      012746  004112      MOV      @FRMT23, -(SP)
26     010000      012746  000004      MOV      @4, -(SP)
27     010004      010600      MOV      SP, R0
28     010006      104414      TRAP    C#PNTB
29     010010      062706  000012      ADD     @12, SP
30
31     010014      013746  031416      ;PRINT 'RECVD: XXXXXX XXXXXX'
32     010020      013746  031414      MOV      TDWRD2, -(SP)
33     010024      012746  006015      MOV      TDWRD1, -(SP)
34     010030      012746  004112      MOV      @RCVD, -(SP)
35     010034      012746  000004      MOV      @FRMT23, -(SP)
36     010040      010600      MOV      @4, -(SP)
37     010042      104414      MOV      SP, R0
38     010044      062706  000012      TRAP    C#PNTB
39     010050      004737  011132      ADD     @12, SP
40
41     010054      012746  002656      JSR      PC, DMPREG      ;DUMP THE ERROR REGISTERS
42     010060      012746  000001      ;CR-LF
43     010064      010600      MOV      @CRLF, -(SP)
44     010066      104414      MOV      @1, -(SP)
45     010070      062706  000004      MOV      SP, R0
46
47     010074      104423      L10003: TRAP    C#MSG
48
49     010076      004737  011046      ERR2::     JSR      PC, TYPLOC      ;REPORT THE ADDRESS OF THE FAILURE
50     010102      012746  003661      MOV      @FRMT16, -(SP)      ;PRINT 'DRIVE WORD#1 WORD#2 WORD#3 WORD#4 WORD#5 WORD#6'
51     010106      012746  000001      MOV      @1, -(SP)
52     010112      010600      MOV      SP, R0
53     010114      104414      TRAP    C#PNTB
    
```

	010116	062706	000004		ADD	#4, SP	
31	010122	013746	031426		MOV	TDWRD6, -(SP)	
	010126	013746	031424		MOV	TDWRD5, -(SP)	
	010132	013746	031422		MOV	TDWRD4, -(SP)	
	010136	013746	031420		MOV	TDWRD3, -(SP)	
	010142	013746	031416		MOV	TDWRD2, -(SP)	
	010146	013746	031414		MOV	TDWRD1, -(SP)	
	010152	013746	002434		MOV	TRVNO, -(SP)	
	010156	012746	003754		MOV	#FRMT17, -(SP)	
	010162	012746	000010		MOV	#10, -(SP)	
	010166	010600			MOV	SP, RO	
	010170	104414			TRAP	C#PNTB	
	010172	062706	000022		ADD	#22, SP	
32							;CR-LF
33	010176	012746	002656		MOV	#CRLF, -(SP)	
	010202	012746	000001		MOV	#1, -(SP)	
	010206	010600			MOV	SP, RO	
	010210	104414			TRAP	C#PNTB	
	010212	062706	000004		ADD	#4, SP	
34	010216			L10004:			
	010216	104423			TRAP	C#MSG	
35							
36	010220			ERR3::			
37	010220	004737	011046		JSR	PC, TYPLOC	;REPORT THE ADDRESS OF THE FAILURE
38							;CR-LF
39	010224	012746	002656		MOV	#CRLF, -(SP)	
	010230	012746	000001		MOV	#1, -(SP)	
	010234	010600			MOV	SP, RO	
	010236	104414			TRAP	C#PNTB	
	010240	062706	000004		ADD	#4, SP	
40	010244			L10005:			
	010244	104423			TRAP	C#MSG	
41							
42	010246			ERR5::			
43	010246	004737	011046		JSR	PC, TYPLOC	;REPORT THE ADDRESS OF THE FAILURE
44							;PRINT 'EXPCTD: XXXXXX RECVD: XXXXXX'
45	010252	013746	002300		MOV	RECVD, -(SP)	
	010256	013746	002276		MOV	EXPCTD, -(SP)	
	010262	012746	004351		MOV	#FRMT33, -(SP)	
	010266	012746	000003		MOV	#3, -(SP)	
	010272	010600			MOV	SP, RO	
	010274	104414			TRAP	C#PNTB	
	010276	062706	000010		ADD	#10, SP	
46	010302	004737	011132		JSR	PC, DMPREG	;DUMP THE ERROR REGISTERS
47							;CR-LF
48	010306	012746	002656		MOV	#CRLF, -(SP)	
	010312	012746	000001		MOV	#1, -(SP)	
	010316	010600			MOV	SP, RO	
	010320	104414			TRAP	C#PNTB	
	010322	062706	000004		ADD	#4, SP	
49	010326			L10006:			
	010326	104423			TRAP	C#MSG	
50							

```

1          .SBTTL GLOBAL SUBROUTINES SECTION
2
3          ;AUTO SIZE FOR RH70 CONTROLLER AND DETERMINE IF IT IS JUMPERED FOR 22 OR
4          ;32 REGISTERS
5          ;CALL
6          ;      JSR      PC,SIZE70      ;CALL ROUTINE
7          ;
8          ;R5 MUST CONTAIN POINTER TO NEW RPCS1 BASE ADDRESS
9
11 010330 005037 002430      SIZE70: CLR      RHEXT      ;CLEAR RPBAE OFFSET
12 010334 005037 002432      CLR      RHTYPE     ;CLEAR RHXX TYPE REGISTER (RH11)
13 010340 013746 000004      MOV      ERRVEC, -(SP) ;SAVE CONTENTS OF ERROR VECTOR
14 010344 012737 010414 000004  MOV      #2#,ERRVEC   ;SETUP 'TRAP' RETURN ADDRESS
15 010352 011500      MOV      (R5),R0      ;GET RPCS1 ADDRESS
16 010354 062700 000050      ADD      #50,R0       ;GET REGISTER OFFSET FOR RH70
17 010360 012702 000012      MOV      #10.,R2     ;GET NUMBER OF REGISTERS TO CHECK
18 010364 005720      TST      (R0)+       ;TRAP IF NOT A VALID RPBAE
19 010366 005720      TST      (R0)+       ;TRAP IF NOT A VALID RPCS3
20 010370 012737 000050 002430  MOV      #50,RHEXT    ;LOAD OFFSET FOR RPBAE (22 REGISTER RH)
21 010376 005720      1$: TST      (R0)+       ;TRAP IF NOT A VALID REGISTER
22 010400 005302      DEC      R2          ;DONE WITH ALL 32 REGISTERS ?
23 010402 001375      BNE      1$         ;BR IF NO
24 010404 012737 000074 002430  MOV      #74,RHEXT   ;LOAD OFFSET FOR RPBAE (32 REGISTER RH)
25 010412 000403      BR      3$         ;
26 010414 012716 010422      2$: MOV      #3#,(SP)    ;SETUP RETURN ADDRESS
27 010420 000002      RTI
28
29 010422 011500      3$: MOV      (R5),R0      ;GET RPCS1 REGISTER
30 010424 013702 002430      MOV      RHEXT,R2   ;GET RPBAE REGISTER OFFSET
31 010430 001415      BEQ      4$         ;BR IF NONE
32 010432 060002      ADD      R0,R2      ;GET RPBAE REGISTER
33 010434 052710 001400      BIS      #A17!A16,(R0) ;SET EXTENDED ADDRESS BITS IN RPCS1
34 010440 022712 000003      CMP      #3,(R2)    ;ARE THE EXTENDED BITS SET IN RPBAE ?
35 010444 001007      BNE      4$         ;BR IF NO
36 010446 005012      CLR      (R2)       ;CLEAR EXTENDED ADDRESS BITS IN RPBAE
37 010450 011046      MOV      (R0),-(SP) ;SAVE RPCS1 REG CONTENTS
38 010452 042726 176377      BIC      #+C<A17!A16>,(SP)+ ;ARE THE EXTEND BITS CLEAR IN RPCS1 ?
39 010456 001002      BNE      4$         ;BR IF NO
40 010460 005237 002432      INC      RHTYPE     ;SET RHXX TYPE REGISTER (RH70)
41 010464 012637 000004      4$: MOV      (SP)+,ERRVEC ;RESTORE CONTENTS OF ERROR VECTOR
42 010470 000207      RTS      PC

```

```

1      .SBTTL  DISK DRIVER
2
3      ;*****
4      ;THE 'READY' ROUTINE LOOKS AT REGISTER RPDS TO DETERMINE DRIVE READY STATUS
5      ;AND IS CALLED FROM THE 'DRIVER' ROUTINE.
6      ;THE 'DRIVER' ROUTINE SENDS THE COMMAND TO THE CONTROLLER AND IS CALLED BY: THE
7      ;READ/WRITE FUNCTIONS.
8      ;(INPUTS AND OUTPUTS ARE FOR 'DRIVER' MODULE)
9      ;*
10     ;INPUTS ARE:  NEGWRD, DESCYL, DESTRK, BUSADR, OFFSET, FUNCTN, RPDS, NOWRCK,
11     ;              RPCS1
12     ;
13     ;NEGWRD = WORD COUNT TO BE NEGATED. (GETS NEGATED IN DRIVER)
14     ;DESCYL = DESIRED CYLINDER ADDRESS.
15     ;DESTRK = DESIRED TRACK/SECTOR ADDRESS.
16     ;BUSADR = BUS ADDRESS FOR DATA TRANSFERS.
17     ;OFFSET = CONTENTS OF THE OFFSET REGISTER.
18     ;FUNCTN = COMMAND (FUNCTION) TO BE EXECUTED.
19     ;
20     ;OUTPUTS ARE:  RPWC, RPDA, RPCS1, RPBA, ERRMSK
21     ;*
22     ;CALL
23     ;      JSR      PC,DRIVER      ;SEND COMMAND TO CONTROLLER
24     ;*****
25
26 010472 105777 171754  READY:  TSTB  @RPDS      ;IS DRIVE READY SET ?
27 010476 100375          BPL      READY      ;BR IF NO
28 010500 000207          RTS      PC
29
30 010502 005037 002272  DRIVER:  CLR      ERRMSK      ;RESET NO ERROR STATUS
31 010506 005437 002376          NEG      NEGWRD      ;NEGATE THE WORD COUNT ONLY ONCE!
32 010512 032777 040000 171732 1$:  BIT      @ERR,@RPDS    ;DO WE HAVE A COMPOSITE ERROR?
33 010520 001134          BNE      11$      ;BR IF YES
34
35 010522 004737 010472 2$:  JSR      PC,READY      ;WAIT FOR DRIVE READY
36 010526 013777 002376 171706  MOV      NEGWRD,@RPWC  ;LOAD WORD COUNT,
37 010534 013777 002312 171732  MOV      DESCYL,@RPDC  ;CYLINDER,
38 010542 013777 002314 171676  MOV      DESTRK,@RPDA  ;TRK/SEC,
39 010550 013777 002234 171666  MOV      BUSADR,@RPBA  ;BUS ADDRESS,
40 010556 013777 002236 171706 3$:  MOV      OFFSET,@RPOF  ;AND OFFSET REGISTERS.
41 010564 042777 100000 171672  BIC      @DMD,@RPMR1   ;ASSUME NOT DIAGNOSTIC MODE
42 010572 013746 002222          MOV      LSCYL,-(SP)   ;GET LAST USER CYLINDER ADDRESS AND
43 010576 005216          INC      (SP)        ;MAKE THE FIRST FE CYLINDER.
44 010600 023726 002312          CMP      DESCYL,(SP)+ ;ACCESSING FE CYLINDERS ?
45 010604 103403          BLO      4$          ;IF NOT, SKIP NEXT INSTRUCTION
46 010606 052777 100000 171650  BIS      @DMD,@RPMR1  ;SET DIAGNOSTIC MODE
47 010614 013777 002404 171616 4$:  MOV      FUNCTN,@RPCS1 ;GET THE COMMAND TO EXECUTE
48 010622 017746 171612 5$:  MOV      @RPCS1,-(SP) ;GET STATUS
49 010626 042716 177576          BIC      @+C<RDY!GO>,(SP)
50 010632 022726 000200          CMP      @RDY,(SP)+  ;RDY=1,GO=0 ?
51 010636 001371          BNE      5$          ;BR IF NO
52
53     ;FUNCTION COMPLETE, LOOK FOR ERRORS
54
55 010640 032777 040000 171572  BIT      @TRE,@RPCS1  ;IS TRE SET ?
56 010646 001433          BEQ      8$          ;BR IF NO
57 010650 032777 100000 171624  BIT      @BSE,@RPER3  ;SEE IF A BSE WAS DETECTED DURING THE

```

```

58 010656 001413          BEQ      6#          ;DATA TRANSFER AND BRANCH IF NOT, ELSE
59 010660 012777 000011 171552  MOV     #DRVCLR,#RPCS1 ;DO A DRIVE CLEAR.
60 010666 005777 171550          TST     #RPWC         ;IF THE DATA TRANSFER WAS COMPLETED,
61 010672 001452          BEQ      12#         ;THEN EXIT DRIVER NOW.
62 010674 022737 000073 002404  CMP     #RDHD,FUNCTN ;SEE IF COMMAND WAS A READ HEADER, ELSE
63 010702 001046          BNE     12#         ;EXIT DRIVER IF NOT.
64 010704 000724          BR      3#          ;CONTINUE COMMAND...
65
66 010706 032777 040000 171534 6# :  BIT     #WCE,#RPCS2   ;IS 'WCE' SET ?
67 010714 001404          BEQ      7#          ;BR IF NO
68 010716 012737 000007 002272  MOV     #7,ERRMSK    ;LOG 'WRITE CHECK ERROR' STATUS
69 010724 000435          BR      12#         ;
70 010726 012737 000014 002272 7# :  MOV     #14,ERRMSK   ;LOG 'TRANSFER ERROR' STATUS
71 010734 000431          BR      12#         ;
72
73 010736 023727 002404 000035 8# :  CMP     FUNCTN,#DIAG ;WAS IT A "HOUSEKEEP" COMMAND ?
74 010744 101416          BLOS   10#         ;BR IF NO
75 010746 005777 171500          TST     #RPDS         ;DID WE GET AN UNEXPECTED ATA ?
76 010752 100004          BPL     9#          ;BR IF NO
77 010754 012737 000006 002272  MOV     #6,ERRMSK    ;LOG 'UNEXPECTED ATTENTION ERROR' STATUS
78 010762 000416          BR      12#         ;
79
80 010764 005777 171450          TST     #RPCS1        ;IS SPECIAL CONDITION SET ?
81 010770 100004          BPL     10#         ;BR IF NO
82 010772 012737 000015 002272  MOV     #15,ERRMSK   ;LOG 'SPECIAL CONDITION ERROR' STATUS
83 011000 000407          BR      12#         ;
84
85 011002 032777 040000 171442 10# :  BIT     #ERR,#RPDS   ;DID WE GET A COMPOSITE ERROR?
86 011010 001404          BEQ     13#         ;BR IF NO
87 011012 012737 000002 002272 11# :  MOV     #2,ERRMSK    ;LOG 'COMPOSITE ERROR' STATUS
88 011020 000207          RTS     PC          ;EXIT
89
90 011022 005737 002176          TST     NOWRCK        ;INHIBIT WRITE CHECK OPERATION ?
91 011026 003374          BGT     12#         ;BR IF YES
92 011030 022737 000063 002404  CMP     #FMTRK,FUNCTN ;WAS THE LAST FUNCTION A FORMAT TRACK ?
93 011036 001370          BNE     12#         ;BR IF NO
94 011040 004737 011734          JSR     PC,WRITCK    ;DO THE WRITE-CHECK OPERATION
95 011044 000616          BR      DRIVER

```



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15 011046
16 011046 013746 002260
   011052 012746 004070
   011056 012746 000002
   011062 010600
   011064 104414
   011066 062706 000006
17
18 011072 113737 002315 002362
19
20 011100 013746 002362
   011104 013746 002312
   011110 012746 004034
   011114 012746 000003
   011120 010600
   011122 104414
   011124 062706 000010
21 011130 000207

```

```

;*****
;THIS MODULE IS USED BY THE ERROR REPORTING SECTION OF THE PROGRAM. THE
;PURPOSE OF THIS MODULE IS TO REPORT THE FUNCTION IN PROCESS AT THE TIME OF
;FAILURE, THE TRACK AND CYLINDER IN PROCESS AT THE TIME OF THE FAILURE.
;*
;INPUTS ARE:  ERRFNC, DESTRK+1, DESCYL
;
;OUTPUT IS USER CONSUMABLE.
;*
;CALL
;      JSR      PC,TYPLOC
;*****
TYPLOC:                                ;PRINT 'PRSNT FUNCT: '
      MOV      ERRFNC,-(SP)
      MOV      @FRMT22,-(SP)
      MOV      #2,-(SP)
      MOV      SP,R0
      TRAP    C$PNTB
      ADD     #6,SP
      MOVB    DESTRK+1,CONTLT ;GET THE TRACK ADDRESS
                                ;PRINT 'CYL:   TRK: '
      MOV      CONTLT,-(SP)
      MOV      DESCYL,-(SP)
      MOV      @FRMT20,-(SP)
      MOV      #3,-(SP)
      MOV      SP,R0
      TRAP    C$PNTB
      ADD     #10,SP
      RTS     PC                ;RETURN

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

```
*****
;THIS MODULE IS USED BY THE ERROR REPORTING SECTION OF THE PROGRAM. IT
;PRODUCES A REGISTER SNAPSHOT OF THE RHXX REGISTERS SUBSEQUENT TO A DETECTABLE
;ERROR.
;*
;INPUTS ARE: FROM REG - REG+52
;
;THE OUTPUT IS USER CONSUMABLE.
;*
;CALL
; JSR PC,DMPREG
;*****
```

```
011132
011132 012746 002664
011136 012746 000001
011142 010600
011144 104415
011146 062706 000004
011152 013746 002526
011156 013746 002524
011162 013746 002522
011166 013746 002520
011172 013746 002516
011176 013746 002514
011202 013746 002434
011206 012746 002755
011212 012746 000010
011216 010600
011220 104415
011222 062706 000022
011226 012746 003035
011232 012746 000001
011236 010600
011240 104415
011242 062706 000004
011246 013746 002544
011252 013746 002542
011256 013746 002540
011262 013746 002536
011266 013746 002534
011272 013746 002532
011276 013746 002530
011302 012746 003126
011306 012746 000010
011312 010600
011314 104415
011316 062706 000022
011322 012746 003206
011326 012746 000001
011332 010600
011334 104415
011336 062706 000004
011342 013746 002562
```

```
DMPREG:
MOV #FRMT00,-(SP) ;PRINT 'DRIVE RPCS1 RPWC RPBA RPDA RPCS2 RPDS'
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV REG+12,-(SP)
MOV REG+10,-(SP)
MOV REG+6,-(SP)
MOV REG+4,-(SP)
MOV REG+2,-(SP)
MOV REG,-(SP)
MOV DRVNO,-(SP)
MOV #FRMT01,-(SP)
MOV #10,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #22,SP
;PRINT 'RPER1 RPAS RPLA RPDB RPMR1 RPDT RPSN'
MOV #FRMT02,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV REG+30,-(SP)
MOV REG+26,-(SP)
MOV REG+24,-(SP)
MOV REG+22,-(SP)
MOV REG+20,-(SP)
MOV REG+16,-(SP)
MOV REG+14,-(SP)
MOV #FRMT03,-(SP)
MOV #10,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #22,SP
;PRINT 'RPOF RPDC RPCC RPER2 RPER3 RPEC1 RPEC2'
MOV #FRMT04,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #4,SP
MOV REG+46,-(SP)
```

011346	013746	002560	MOV	REG+44,-(SP)	
011352	013746	002556	MOV	REG+42,-(SP)	
011356	013746	002554	MOV	REG+40,-(SP)	
011362	013746	002552	MOV	REG+36,-(SP)	
011366	013746	002550	MOV	REG+34,-(SP)	
011372	013746	002546	MOV	REG+32,-(SP)	
011376	012746	003300	MOV	#FRMT05,-(SP)	
011402	012746	000010	MOV	#10,-(SP)	
011406	010600		MOV	SP,R0	
011410	104415		TRAP	C#PNTX	
011412	062706	000022	ADD	#22,SP	
24					
25	011416	005737	TST	RHTYPE	;IS IT RH70 CONTROLLER ?
26	011422	001424	BEG	1#	;BR IF NO
27					;PRINT 'RPBAE RPCS3'
28	011424	012746	MOV	#FRMT06,-(SP)	
	011430	012746	MOV	#1,-(SP)	
	011434	010600	MOV	SP,R0	
	011436	104415	TRAP	C#PNTX	
	011440	062706	ADD	#4,SP	
29	011444	013746	MOV	REG+52,-(SP)	
	011450	013746	MOV	REG+50,-(SP)	
	011454	012746	MOV	#FRMT07,-(SP)	
	011460	012746	MOV	#3,-(SP)	
	011464	010600	MOV	SP,R0	
	011466	104415	TRAP	C#PNTX	
	011470	062706	ADD	#10,SP	
30	011474	000207	RTS	PC	;RETURN

1#:

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

;*****
;THIS MODULE IS USED TO LOG DATA INTO THE BAD SECTOR FILE.  THIS DATA IS PUT
;INTO THE BSF BECAUSE IT WON'T FIT INTO THE TD FOR AN EXISTING TRACK.
;*
;INPUTS ARE:  FROM TMPBSF, TMPBSF+2
;
;OUTPUTS ARE: TO A BUFFER CALLED 'MFGBUF' (OUTPUT IS CONTROLLED BY 'ENTBSF',
;              INDICATING THAT THE BSF IS ALREADY FULL)
;*
;CALL
;      JSR      PC,LODBSF      ;CALL ROUTINE
;*****

```

```

LODBSF:
      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
      TST     MODTD        ;SEE IF MANUAL MODIFY IS ENABLED.
      BEQ     1$           ;BRANCH IF NOT, ELSE
      TST     MODBY        ;CHECK TO SEE IF MODIFIING BY WORDS OR INDEX.
      BEQ     2$           ;BRANCH IF WORD COUNT MODE, ELSE
      BR      5$           ;MUST BE INDEX MODE. SO EXIT.

1$:   TST     ENWTTD        ;SEE IF WRT TD IS ENABLED DURING SCAN MODE.
      BEQ     5$           ;BRANCH IF NOT, ELSE
2$:   MOV     ENTBSF,R2     ;GET POINTER FOR ENTRY INTO BSF
      CMP     R2,#ENDBSF   ;IS BSF BUFFER FULL ?
      BHI     4$           ;BR IF YES
      CMP     TMPBSF,LSCYL ;SEE WHAT CYLINDER ADDRESS IS BEING FLAGGED.
      BHI     5$           ;BRANCH IF FE CYLINDER(S).
      BLO     3$           ;BRANCH IF ANY USER CYLINDER, ELSE
      CMPB   TMPBSF+3,LSTRK ;SEE IF WE ARE FLAGGING THE BSF AREA.
      BEQ     5$           ;BRANCH IF SO, ELSE
3$:   MOV     TMPBSF,(R2)+  ;LOG THE BSF DATA.
      MOV     TMPBSF+2,(R2)+ ;CYLINDER/TRACK/SECTOR
      MOV     R2,ENTBSF    ;UPDATE POINTER FOR NEXT ENTRY INTO BSF
      BR      5$           ;AND FOR NOW, TAKE THE RETURN
4$:   PRINT  'WARNING ** BAD SECTOR FILE BUFFER IS
;           FULL, NO MORE ENTRIES ALLOWED'

      MOV     #WARN,-(SP)
      MOV     #FRMT63,-(SP)
      MOV     #2,-(SP)
      MOV     SP,R0
      TRAP   C#PNTF
      ADD    #6,SP

5$:   MOV     (SP)+,R2     ;;POP STACK INTO R2
      RTS     PC          ;RETURN TO THE CALLER

ENTBSF: .WORD 0          ;CONTAINS ADDRESS POINTER FOR NEXT ENTRY INTO
;THE BAD SECTOR FILE.

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 011630
    011630 010146
    011632 010246
    011634 010346
    011636 010446
    011640 010546
17 011642 012701 032464
18 011646 012702 034744
19 011652 012703 000002
20 011656 113705 002322
21 011660 005205
22 011664 006205
23 011666 012704 000006
24 011672 012122
25 011674 005304
26 011676 003375
27 011700 062701 000014
28 011704 005305
29 011706 003367
30 011710 012701 032500
31 011714 005303
32 011716 003357
33 011720 012605
    011722 012604
    011724 012603
    011726 012602
    011730 012601
34 011732 000207

;*****
;THIS MODULE DOES THE BUFFER CHANGING WHEN A WRITE-CHECK OR READ HEADER & DATA
;OPERATION IS REQUIRED ON A DRIVE WHICH WAS FORMATTED IN INTERLEAVED MODE.
;THE WRITE CHECK AND READ HEADER & DATA FUNCTION ARE ALWAYS DONE IN A LINEAR
;FASHION, IE: SECTOR (N), SECTOR (N+1), SECTOR (N+2), SECTOR (N+...),
;REGARDLESS OF WHETHER THE DRIVE IS IN INTERLEAVED OR NON-INTERLEAVED MODE.
;*
;INPUTS ARE:   HDRBLK
;
;OUTPUTS ARE:  LINBUF
;*
;THIS MODULE IS CALLED BY: WRITCK, VERIFY
;*****

CHABUF:
MOV     R1,-(SP)      ;;PUSH R1 ON STACK
MOV     R2,-(SP)      ;;PUSH R2 ON STACK
MOV     R3,-(SP)      ;;PUSH R3 ON STACK
MOV     R4,-(SP)      ;;PUSH R4 ON STACK
MOV     R5,-(SP)      ;;PUSH R5 ON STACK
MOV     @HDRBLK,R1    ;GET POINTER TO BEGINNING OF INTERLEAVED DATA
MOV     @LINBUF,R2    ;GET POINTER TO BEGINNING OF LINEAR BUFFER
MOV     @2,R3         ;AND DO FOR 2 ITERATIONS
1$:     MOV     SEC50,R5 ;GET THE # OF SECTORS
        INC     R5     ;+1
        ASR     R5     ;AND DIVIDE BY 2
2$:     MOV     @6,R4   ;# OF WORDS/SECTOR
3$:     MOV     (R1)+,(R2)+ ;LOG THE DATA
        DEC     R4     ;REDUCE # OF REMAINING WORDS
        BGT     3$     ;IF > 0, DO AGAIN
        ADD     @12.,R1 ;SKIP A SECTOR
        DEC     R5     ;ONE LESS SECTOR TO DO
        BGT     2$     ;IF > 0, KEEP ON GOING
        MOV     @HDRBLK+12.,R1 ;GET SECOND HALF OF BUFFER
        DEC     R3     ;SECOND PASS, R3=1
        BGT     1$     ;IF > 0, IT TIME FOR SECOND PASS
        MOV     (SP)+,R5 ;;POP STACK INTO R5
        MOV     (SP)+,R4 ;;POP STACK INTO R4
        MOV     (SP)+,R3 ;;POP STACK INTO R3
        MOV     (SP)+,R2 ;;POP STACK INTO R2
        MOV     (SP)+,R1 ;;POP STACK INTO R1
        RTS     PC     ;RETURN

```

```

1      .SBTTL  WRITE BUFFER POSTFIX MODULE
2
3      ;;*****
4      ;THIS MODULE PERFORMS THE BUFFER POSTFIXING WHEN AN INTERLEAVED FORMAT
5      ;OPERATION WAS PERFORMED.  THE WRITE CHECK OPERATION IS PERFORMED IN A LINEAR
6      ;FASHION, IE: SECTOR (N), SECTOR (N+1), SECTOR (N+2), SECTOR (N+...).
7      ;*
8      ;INPUTS ARE:  INTLEV, LINBUF, WCKHD, WCKHDR, TKWCNT
9      ;
10     ;OUTPUTS ARE:  BUSADR, FUNCTN, ERRFNC, RPOF, NEGWRD
11     ;*
12     ;THIS ROUTINE IS CALLED FROM THE 'DRIVER'.
13     ;;*****
14
15 011734 105737 002244  WRITCK: TSTB  INTLEV      ;INTERLEAVE MODE ?
16 011740 001405          BEQ      1$      ;BR IF NO
17 011742 004737 011630  JSR      PC,CHABUF ;RE-SHUFFLE INTERLEAVED DATA AND
18 011746 012737 034744 002234  MOV     @LINBUF,BUSADR ;GET POINTER TO BEGINNING OF LINEAR BUFFER
19 011754 012737 000053 002404 1$: MOV   @WCKHD,FUNCTN ;SETUP A WRITE-CHECK HEADERS OPERATION
20 011762 012737 005653 002260  MOV     @WCKHDR,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
21 011770 052737 100000 002236  BIS    @CMOD,OFFSET  ;SET COMMAND MODIFIER BIT
22 011776 013737 002364 002376  MOV     TKWCNT,NEGWRD ;AND SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
23 012004 000207          RTS      PC      ;NOW TAKE THE RETURN

```

```

1      .SBTTL  MANUAL INPUT ROUTINE
2
3      ;*****
4      ;THIS MODULE COLLECTS THE USER INPUT TO MODIFY AN EXISTING TRACK DESCRIPTOR IT
5      ;THEN TAKES THE USER VALUES AND MERGES THEM INTO AN EXISTING OR NEW TRACK
6      ;DESCRIPTOR.
7      ;*
8      ;INPUTS ARE:  TDCPY3 - TDCPY6, DESCYL, DESTRK, #NULL
9      ;
10     ;OUTPUTS ARE: TDCPY3 - TDCPY6, TDWRD1 - TDWRD6, TEXT, RELWRD, SECADD
11     ;*
12     ;THIS MODULE IS CALLED BY: MAIN
13     ;*****
14
15     012006      INPUTD:
16     012006      010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
17     012010      010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
18                                     ;;PRINT 'MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE)?'
19     012012      104443      TRAP     C#GMAN
20     012014      000406      BR      10000#
21     012016      002202      .WORD  MODBY
22     012020      000052      .WORD  T#CODE
23     012022      006570      .WORD  MSG22
24     012024      000003      .WORD  3
25     012026      000000      .WORD  T#LOLIM
26     012030      000002      .WORD  T#HILIM
27     012032      005737      10000#:  TST     ANYWHR      ;DON'T ALLOW DESIRED CYLINDER TO BE CHANGED.
28     012036      001410      BEQ     1#          ;IF WRITTING ON FE CYLINDER ONLY, ELSE
29                                     ;PRINT 'CYLINDER ADDRESS (D) 0 ?'
30     012040      104443      TRAP     C#GMAN
31     012042      000406      BR      10001#
32     012044      002312      .WORD  DESCYL
33     012046      000052      .WORD  T#CODE
34     012050      012654      .WORD  MSG38
35     012052      177777      .WORD  177777
36     012054      000000      .WORD  T#LOLIM
37     012056      001166      .WORD  T#HILIM
38     012060      10001#:  1#:
39                                     ;PRINT 'TRACK ADDRESS (D) 0 ?'
40     012060      104443      TRAP     C#GMAN
41     012062      000406      BR      10002#
42     012064      002314      .WORD  DESTRK
43     012066      000052      .WORD  T#CODE
44     012070      012675      .WORD  MSG39
45     012072      177400      .WORD  177400
46     012074      000000      .WORD  T#LOLIM
47     012076      000037      .WORD  T#HILIM
48     012100      10002#:
49
50     012100      004737      021440      JSR     PC,READTD  ;CALL THE READ TD SUBROUTINE
51     012104      000401      BR      2#        ;RETURN HERE IF EXCEEDED RETRY LIMIT
52     012106      000402      BR      3#        ;ELSE RETURN HERE, NO ERROR
53     012110      004737      026332      2#:  JSR     PC,DESTD ;GET DESIRED TD, BECAUSE WE FAILED THE READ
54
55     012114      012737      006030      3#:  MOV     #CURRENT,TEXT ;LOAD TEXT WITH ADDRESS OF MESSAGE
56     012122      004737      017240      JSR     PC,TDDUMP  ;GO DUMP THE TRACK DESCRIPTOR
    
```

```

34 012126 023727 002202 000001 4: CMP MODBY,#1 ;MODIFY BY WORDS, INDEX OR CHANGE MODE ?
35 012134 002512 BLT 6: ;BR IF WORDS
36 012136 001545 BEQ 7: ;BR IF INDEX
37 ;MUST BE CHANGE MODE
38 012140 042737 140000 031414 BIC #BIT14!BIT15,TDWRD1 ;RESET TD & HDR 0 MOVED BITS
39 012146 005037 031434 CLR TDCPY3 ;PUT DUMMY NULL IN TD WORD #3
45 012152 005037 031436 CLR TDCPY4 ;PUT DUMMY NULL IN TD WORD #4
012156 005037 031440 CLR TDCPY5 ;PUT DUMMY NULL IN TD WORD #5
012162 005037 031442 CLR TDCPY6 ;PUT DUMMY NULL IN TD WORD #6
49 ;PRINT 'TYPE <CR> TO INPUT (0=140000) DATA;'
50 012166 012746 004133 MOV #FRMT24,-(SP)
012172 012746 000001 MOV #1,-(SP)
012176 010600 MOV SP,R0
012200 104414 TRAP C#PNTB
012202 062706 000004 ADD #4,SP
51 ;PRINT 'TD WORD#3 (D) 0 ?'
52 012206 104443 TRAP C#GMAN
012210 000406 BR 10003:
012212 031434 .WORD TDCPY3
012214 000052 .WORD T#CODE
012216 012772 .WORD MSG43
012220 177777 .WORD 177777
012222 000166 .WORD T#LOLIM
012224 042077 .WORD T#HILIM
012226 10003:
53 012226 005737 031434 TST TDCPY3 ;NULL ENTRY ?
54 012232 001436 BEQ 5: ;BR IF YES
55 ;PRINT 'TD WORD#4 (D) 0 ?'
56 012234 104443 TRAP C#GMAN
012236 000406 BR 10004:
012240 031436 .WORD TDCPY4
012242 000052 .WORD T#CODE
012244 013004 .WORD MSG44
012246 177777 .WORD 177777
012250 000001 .WORD T#LOLIM
012252 042077 .WORD T#HILIM
012254 10004:
57 012254 005737 031436 TST TDCPY4 ;NULL ENTRY ?
58 012260 001423 BEQ 5: ;BR IF YES
59 ;PRINT 'TD WORD#5 (D) 0 ?'
60 012262 104443 TRAP C#GMAN
012264 000406 BR 10005:
012266 031440 .WORD TDCPY5
012270 000052 .WORD T#CODE
012272 013016 .WORD MSG45
012274 177777 .WORD 177777
012276 000001 .WORD T#LOLIM
012300 042077 .WORD T#HILIM
012302 10005:
61 012302 005737 031440 TST TDCPY5 ;NULL ENTRY ?
62 012306 001410 BEQ 5: ;BR IF YES
63 ;PRINT 'TD WORD#6 (D) 0 ?'
64 012310 104443 TRAP C#GMAN
012312 000406 BR 10006:
012314 031442 .WORD TDCPY6
012316 000052 .WORD T#CODE
012320 013030 .WORD MSG46

```



```

012322 177777      .WORD 177777
012324 000001      .WORD T%LOLIM
012326 042077      .WORD T%HILIM
012330
65 012330 013737 031434 031420 5%: MOV TDCPY3,TDWRD3 ;GET TD WORD #3
66 012336 013737 031436 031422 5%: MOV TDCPY4,TDWRD4 ;GET TD WORD #4
67 012344 013737 031440 031424 5%: MOV TDCPY5,TDWRD5 ;GET TD WORD #5
68 012352 013737 031442 031426 5%: MOV TDCPY6,TDWRD6 ;GET TD WORD #6
69 012360 000454 5%: BR 9%
70 012362 6%: ;PRINT 'SECTOR ADDRESS (D) ?'
71 012362 104443 TRAP C%GMAN
012364 000406 BR 10007%
012366 002402 .WORD SECADD
012370 000042 .WORD T%CODE
012372 012753 .WORD MMSG42
012374 177777 .WORD 177777
012376 000000 .WORD T%LOLIM
012400 000061 .WORD T%HILIM
012402 10007%:
72 ;PRINT 'WORD IN SECTOR (D) ?'
73 012402 104443 TRAP C%GMAN
012404 000406 BR 10010%
012406 002400 .WORD RELWRD
012410 000042 .WORD T%CODE
012412 012713 .WORD MMSG40
012414 177777 .WORD 177777
012416 000000 .WORD T%LOLIM
012420 000401 .WORD T%HILIM
012422 10010%:
74 012422 004737 016536 JSR PC,RELTIME ;CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
75 012426 023737 002400 002354 CMP RELWRD,TDLMTS+2 ;WILL BEGINNING OF DEFECT BE IN GAP 2 ?
76 012434 003016 BGT 8% ;BR IF NO
77 012436 013737 002354 002400 MOV TDLMTS+2,RELWRD ;GET UPPER LIMIT FOR MDR 0 MOVED AND
78 012444 005237 002400 INC RELWRD ;ADD ONE TO IT.
79 012450 000410 BR 8% ;AND GO INSERT IT INTO TD.
80 012452 7%: ;PRINT 'WORDS FROM INDEX (D) ?'
81 012452 104443 TRAP C%GMAN
012454 000406 BR 10011%
012456 002400 .WORD RELWRD
012460 000042 .WORD T%CODE
012462 012732 .WORD MMSG41
012464 177777 .WORD 177777
012466 000166 .WORD T%LOLIM
012470 042553 .WORD T%HILIM
012472 10011%:
82 012472 004737 016616 JSR PC,INSERT ;INSERT THE DEFECT IN TD
83 012476 005737 031460 TST TMPBSF ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
84 012502 100403 BMI 9% ;BR IF YES
85 012504 004737 011476 JSR PC,LOADBSF ;LOAD THE BSF BUFFER.
86 012510 000446 BR 16%
87
88 012512 012700 031420 9%: MOV @TDWRD3,R0 ;GET POINTER TO BEGINNING OF TD BUFFER
89 012516 012701 000004 MOV #4,R1 ;# OF DEFECT WORDS IN TD
90 012522 005720 10%: TST (R0)+ ;IS THIS A DUMMY (0) ENTRY ?
91 012524 001003 BNE 11% ;BR IF NO
92 012526 012760 140000 177776 MOV #NULL,-2(R0) ;MAKE DUMMY (0) ENTRY A NULL (140000) ENTRY
93 012534 005301 11%: DEC R1 ;DONE ALL WORDS YET ?
    
```

```

94 012536 003371          BGT      10#          ;BR IF NO
95
96 012540 013700 002226    MOV      ERRMAX,RO      ;SETUP RETRY COUNT
97 012544 004737 026236    JSR      PC,SAVETD      ;SAVE TO WRITE BUFFER JUST INCASE THERE
98                                ;IS RETRY ATTEMPT.
99 012550 004737 026274    12#:    JSR      PC,RESTTD      ;NOW RESTORE THE TD BUFFER
100 012554 004737 021152   JSR      PC,WRITTD      ;CALL THE WRITE TD SUBROUTINE
101 012560 000421          BR       15#            ;RETURN HERE IF EXCEEDED RETRY LIMIT
102                                ;ELSE RETURN HERE, NO ERROR
103 012562 012737 177777 002252   MOV      #-1,SUPRSS      ;SUPPRESS THE ERROR MESSAGE OUTPUT
104 012570 005300          13#:    DEC      RO              ;DID WE EXCEED RETRY ?
105 012572 002414          BLT      15#            ;BR IF YES
106 012574 003002          BGT      14#            ;BR IF NO
107 012576 005037 002252   CLR      SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
108 012602 004737 021440   14#:    JSR      PC,READTD      ;CALL THE READ TD SUBROUTINE
109 012606 000770          BR       13#            ;RETURN HERE IF EXCEEDED RETRY LIMIT
110                                ;ELSE RETURN HERE, NO ERROR
111 012610 012737 006170 002360   MOV      #CHANGE,TEXT    ;LOAD TEXT WITH ADDRESS OF MESSAGE
112 012616 004737 017240   JSR      PC,TDDUMP       ;GO DUMP THE TRACK DESCRIPTOR
113 012622 000401          BR       16#            ;ADJUST R5 TO SKIP FORMAT TRACK, ON RETURN
114 012624 005725          15#:    TST      (R5)+          ;ALLOW ERROR MESSAGE TO BE OUTPUT
115 012626 005037 002252   16#:    CLR      SUPRSS          ;:POP STACK INTO R1
116 012632 012601          MOV      (SP)+,R1        ;:POP STACK INTO R0
117 012634 012600          MOV      (SP)+,RO        ;PRINT 'DO YOU WANT TO MODIFY ANYMORE TD'S (L) N ?'
118 012636 104443          TRAP     C#GMAN
119 012640 000404          BR       10012#
120 012642 002416          .WORD   MORETD
121 012644 000130          .WORD   T#CODE
122 012646 013042          .WORD   MESG47
123 012650 000001          .WORD   1
124 012652          10012#:
125 012654 000207          RTS      PC              ;TAKE THE RETURN
126
127 012654          103      131      114  MESG38: .ASCIZ /CYLINDER ADDRESS/
128 012675          124      122      101  MESG39: .ASCIZ /TRACK ADDRESS/
129 012713          127      117      122  MESG40: .ASCIZ /WORD IN SECTOR/
130 012732          127      117      122  MESG41: .ASCIZ /WORDS FROM INDEX/
131 012753          123      105      103  MESG42: .ASCIZ /SECTOR ADDRESS/
132 012772          124      104      040  MESG43: .ASCIZ /TD WORD#3/
133 013004          124      104      040  MESG44: .ASCIZ /TD WORD#4/
134 013016          124      104      040  MESG45: .ASCIZ /TD WORD#5/
135 013030          124      104      040  MESG46: .ASCIZ /TD WORD#6/
136 013042          104      117      040  MESG47: .ASCIZ /DO YOU WANT TO MODIFY ANYMORE TD'S/
137
138                                .EVEN
    
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

.SBTTL PROGRAM UTILITIES

;;*****
;THIS MODULE IS USED TO STALL IN LOOPS WHICH MUST WAIT FOR THE RP07
;MICROPROCESSORS. THE WAIT IS A FIXED TIME PERIOD, AND CANNOT BE CHANGED
;DYNAMICALLY. THERE ARE NO SIDE EFFECTS.
;CALL
; MODULE IS CALLED BY: SEIZE
;;*****

WAIT:

MOV #250.,(PC)+
.WORD 0
MOV L#DLY,(PC)+
.WORD 0
DEC -6(PC)
BNE .-4
DEC -22(PC)
BNE .-20
RTS PC ;NOW RETURN TO MAIN

;;*****
;THIS MODULE CHECKS DRY (RPDS-BIT#7) AGAINST A TIMED VALUE. IF RDY IS NOT
;TRUE WITHIN THIS TIME PERIOD, ALL FURTHER ACTION IS ABORTED.

;*
;INPUTS ARE: DRIVE
;
;OUTPUTS ARE: RPCS2, RPDS, RPAS
;*
;THIS MODULE IS CALLED BY: RESET
;;*****

SEIZE: BIS #CLR,RPAS2 ;CLEAR THE MASSBUS
MOV DRVNO,RPAS2 ;RELOAD THE DRIVE NUMBER
TSTB RPDS ;IS THIS DRIVE SEIZED BY ANOTHER PORT?
BMI 3\$;NO, JUST TAKE RETURN
CLR RPDS ;ISSUE A DRIVE REQUEST
MOV R2,-(SP) ;SAVE R2
MOV #10.,R2 ;LOAD R2 WITH AN OVERAL ITERATION COUNT
1\$: TST RPDS ;NOW WAIT FOR THE OTHER PORT TO RELEASE
BNE 2\$;NOT YET IF RPDS = 0
JSR PC,WAIT
DEC R2 ;AND REDUCE THE ITERATION COUNT
BNE 1\$;LOOK AGAIN FOR DRIVE PRESENT
MOV #3,ERRMSK ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
MOV SUPRSS,-(SP) ;PUSH SUPRSS ON STACK
CLR SUPRSS ;ALLOW ERROR MESSAGE TO BE OUTPUT
JSR PC,SAVRPR ;SAVE RP07 REGISTERS ON ERROR
JSR PC,ERRORS ;AND REPORT THE ERROR
MOV (SP)+,SUPRSS ;POP STACK INTO SUPRSS
2\$: MOVB #377,RPAS ;CLEAR ANY UNWANTED ATTENTION BITS
MOV (SP)+,R2 ;RESTORE R2
3\$: RTS PC ;AND TAKE RETURN

013106 012727 000372
013112 000000
013114 013727 002116
013120 000000
013122 005367 177772
013126 001375
013130 005367 177756
013134 001367
013136 000207

013140 052777 000040 167302
013146 013777 002434 167274
013154 105777 167272
013160 100435
013162 005077 167264
013166 010246
013170 012702 000012
013174 005777 167252
013200 001021
013202 004737 013106
013206 005302
013210 001371
013212 012737 000003 002272
013220 013746 002252
013224 005037 002252
013230 004737 017542
013234 004737 013402
013240 012637 002252

013244 112777 000377 167204
013252 012602
013254 000207

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13 013256 004737 013140  
14 013262 012777 000021 167150  
15 013270 012737 010000 002236  
16 013276 004737 010472  
17 013302 000207  
;*****  
;THIS MODULE, WHEN CALLED DOES A CLEAR TO THE DRIVE IT IS DEPENDENT ON THE  
;MODULE "SEIZE" TO RELOAD THE DRIVE NUMBER AFTER THE RESET.  
;*  
;INPUTS ARE: NONE  
;  
;OUTPUTS ARE: RPCS1, RPOF, RPMR1  
;*  
;THIS MODULE IS CALLED BY: RETRY  
;*****  
RESET: JSR PC,SEIZE ;NOW GET THE DRIVE  
MOV #READIN,@RPCS1 ;DO A READ IN PRESET  
MOV #FMT16,OFFSET ;SET UP FOR 16 BIT WORD  
JSR PC,READY ;WAIT FOR DRIVE READY NOW!  
RTS PC ;NOW RETURN TO MAIN
```

```

1      .SBTTL  RETRY ROUTINE
2
3      ;*****
4      ;THIS ROUTINE MAINTAINS THE RETRY COUNTER.  RETRY MAXIMUM IS 'ERRMAX'.  AN
5      ;ERROR WILL BE REPORTED IF THE MAXIMUM RETRY COUNT IS EXCEEDED.
6      ;*
7      ;INPUTS ARE:  ERRMSK
8
9      ;OUTPUTS ARE:  ERRMSK, RTYCNT, ERRCTL
10     ;*
11     ;CALL
12     ;      JSR      PC,RETRY      ;CALL THE ROUTINE
13     ;      -----      ;RETURN HERE IF EXCEEDED MAX RETRY LIMIT
14     ;      -----      ;ELSE RETURN HERE, IF RTYCNT >= 1 THEN ERROR
15     ;                                  IF RTYCNT = 0 THEN NO ERROR
16     ;*
17     ;THIS MODULE IS MAINLY CALLED BY: MAKTRK, DRVINI, RDBSF, WRTBSF, WRITTD,
18     ;                                READTD, HDSCAN, DASCAN
19     ;*****
20
21 013304 005337 002264  RETRY:  DEC      ITCNT      ;DECREMENT ITERATION FOR THIS COMMAND
22 013310 062716 000002      ADD      #2,(SP)  ;ADJUST FOR GOOD RETURN
23 013314 004737 017542      JSR      PC,SAVRPR ;SAVE RP07 REGISTERS
24 013320 005737 002272      TST      ERRMSK   ;DID WE GET ANY ERRORS ?
25 013324 001414          BEQ      1$      ;BR IF NO
26 013326 005237 002274      INC      RTYCNT   ;INCREMENT THE LOCAL RETRY COUNTER AND
27 013332 005237 002270      INC      ERRCTL  ;THE TOTAL ERROR COUNTER.
28 013336 023737 002274 002226  CMP      RTYCNT,ERRMAX ;DID WE EXCEED THE MAX RETRY LIMIT ON ERROR ?
29 013344 103411          BLO      2$      ;BR IF NO
30 013346 162716 000002      SUB      #2,(SP)  ;ADJUST FOR EXCEEDED RETRY LIMIT RETURN
31 013352 004737 013402      JSR      PC,ERRORS ;REPORT THE ERROR NOW!
32
33 013356 005037 002274  1$:   CLR      RTYCNT   ;RESET RETRY COUNT TO 0
34 013362 005737 002272      TST      ERRMSK   ;DID WE GET AN ERROR ?
35 013366 001404          BEQ      3$      ;BR IF NO
36 013370 005037 002272  2$:   CLR      ERRMSK   ;RESET ERROR STATUS AND
37 013374 004737 013256      JSR      PC,RESET ;RESET THE MASSBUS.
38 013400 000207          RTS      PC      ;EXIT

```

```

1      .SBTTL  ERROR HANDLER
2
3      ;*****
4      ;THIS ROUTINE LOOKS AT ERRMSK, AND DECODES THE BIT MASK BITS 0-15 ARE
5      ;APPLICABLE.
6      ;*
7      ;INPUTS ARE:   ERRMSK
8      ;
9      ;OUTPUTS ARE:  ERRMSK, RTYCNT, ERRCTL
10     ;*
11     ;THIS MODULE IS CALLED BY:  RETRY
12     ;*****
13
14 013402 005737 002272  ERRORS: TST      ERRMSK      ;DID WE GET ANY ERRORS?
15 013406 001430          BEQ      4$          ;IF ERRMSK = 0, NO!
16 013410 005737 002252          TST      SUPRSS      ;ALLOW ERROR MESSAGES ?
17 013414 100425          BMI      4$          ;IF MINUS, NO....
18 013416 010146          MOV      R1,-(SP)      ;;PUSH R1 ON STACK
19 013422 012701 013472          MOV      R2,-(SP)      ;;PUSH R2 ON STACK
20 013426 012702 000001          MOV      #5$,R1      ;GET THE TOP OF THE ERROR DISPATCH FILE
21 013432 023702 002272 1$:  MOV      #1,R2      ;AND SET UP THE ERROR NUMBER MASK.
22 013436 001407          CMP      ERRMSK,R2    ;MATCH?
23 013440 005202          BEQ      2$          ;TAKE BRANCH OF DS - REPORT FIND NOW!
24 013442 062701 000012          INC      R2          ;GET NEXT ERROR POSITION
25 013446 020127 013732          ADD      #12,R1      ;MOVE THE DISPATCH POINTER TO NEXT MESSAGE
26 013452 103767          CMP      R1,#6$      ;ARE WE AT THE END OF ERROR TABLE ?
27 013454 000403          BLO     1$          ;BR IF NO
28
29 013456 005237 002266 2$:  INC      ERTTL      ;ADD ONE TO THE ERROR TOTAL AND
30 013462 000111          JMP      (R1)        ;REPORT THE FINF NOW.
31
32 013464          3$:  MOV      (SP)+,R2      ;;POP STACK INTO R2
33 013466 012602 012601          MOV      (SP)+,R1      ;;POP STACK INTO R1
34 013470 000207          RTS      PC          ;RETURN
35
36 013472          5$:  ERRHRD  16,EM16,ERR3 ;TD NOT READABLE THRU
37 013472 104456          TRAP   C$ERHRD
38 013474 000020          .WORD  16
39 013476 007454          .WORD  EM16
40 013500 010220          .WORD  ERR3
41
42 013502 000770          BR      3$          ;RELOCATION ERROR      ;ERRMSK=1
43
44 013504          ERRHRD  3,EM3,ERRO ;COMPOSITE ERROR      ;ERRMSK=2
45 013504 104456          TRAP   C$ERHRD
46 013506 000003          .WORD  3
47 013510 007037          .WORD  EM3
48 013512 007722          .WORD  ERRO
49 013514 000763          BR      3$          ;RETURN
50
51 013516          ERRHRD  4,EM4,ERRO ;DRIVE HUNG           ;ERRMSK=3
52 013516 104456          TRAP   C$ERHRD
53 013520 000004          .WORD  4
54 013522 007063          .WORD  EM4
55 013524 007722          .WORD  ERRO

```

46	013526	000756	BR	3\$;RETURN	
47						
48	013530		ERRHRD	5,EM5,ERRO	;DRIVE WRITE LOCKED	;ERRMSK=4
	013530	104456	TRAP	C\$ERHRD		
	013532	000005	.WORD	5		
	013534	007123	.WORD	EM5		
	013536	007722	.WORD	ERRO		
49	013540	000751	BR	3\$;RETURN	
50						
51	013542		ERRHRD	6,EM6,ERRO	;DRIVE OFFLINE	;ERRMSK=5
	013542	104456	TRAP	C\$ERHRD		
	013544	000006	.WORD	6		
	013546	007146	.WORD	EM6		
	013550	007722	.WORD	ERRO		
52	013552	000744	BR	3\$;RETURN	
53						
54	013554		ERRHRD	7,EM7,ERRO	;UNEXPECTED ATTN	;ERRMSK=6
	013554	104456	TRAP	C\$ERHRD		
	013556	000007	.WORD	7		
	013560	007164	.WORD	EM7		
	013562	007722	.WORD	ERRO		
55	013564	000737	BR	3\$;RETURN	
56						
57	013566		ERRHRD	10,EM10,ERRO	;WRITE CHECK ERROR	;ERRMSK=7
	013566	104456	TRAP	C\$ERHRD		
	013570	000012	.WORD	10		
	013572	007215	.WORD	EM10		
	013574	007722	.WORD	ERRO		
58	013576	000732	BR	3\$;RETURN	
59						
60	013600		ERRHRD	11,EM11,ERR5	;FORMAT VERIFY ERROR	;ERRMSK=10
	013600	104456	TRAP	C\$ERHRD		
	013602	000013	.WORD	11		
	013604	007243	.WORD	EM11		
	013606	010246	.WORD	ERR5		
61	013610	000725	BR	3\$;RETURN	
62						
63	013612		ERRHRD	1,EM1,ERR1	;TD ADDRESS INCORRECT	;ERRMSK=11
	013612	104456	TRAP	C\$ERHRD		
	013614	000001	.WORD	1		
	013616	006766	.WORD	EM1		
	013620	007754	.WORD	ERR1		
64	013622	000720	BR	3\$;RETURN	
65						
66	013624		ERRHRD	2,EM2,ERR2	;TD FORMAT INCORRECT	;ERRMSK=12
	013624	104456	TRAP	C\$ERHRD		
	013626	000002	.WORD	2		
	013630	007013	.WORD	EM2		
	013632	010076	.WORD	ERR2		
67	013634	000713	BR	3\$;RETURN	
68						
69	013636		ERRHRD	12,EM12,ERRO	;DRY DIDN'T RESET	;ERRMSK=13
	013636	104456	TRAP	C\$ERHRD		
	013640	000014	.WORD	12		
	013642	007267	.WORD	EM12		
	013644	007722	.WORD	ERRO		
70	013646	000706	BR	3\$;RETURN	

```

71
72 013650          ERRHRD  13,EM13,ERRO  ;TRANSFER ERROR      ;ERRMSK=14
    013650 104456  TRAP      C$ERHRD
    013652 000015  .WORD    13
    013654 007326  .WORD    EM13
    013656 007722  .WORD    ERRO
73 013660 000701  BR        3$      ;RETURN
74
75 013662          ERRHRD  14,EM14,ERRO  ;SPECIAL CONDITION   ;ERRMSK=15
    013662 104456  TRAP      C$ERHRD
    013664 000016  .WORD    14
    013666 007351  .WORD    EM14
    013670 007722  .WORD    ERRO
76 013672 000674  BR        3$      ;RETURN
77
78 013674          ERRHRD  15,EM15,ERRO  ;RANDOM HEADER ERRORS ;ERRMSK=16
    013674 104456  TRAP      C$ERHRD
    013676 000017  .WORD    15
    013700 007377  .WORD    EM15
    013702 007722  .WORD    ERRO
79 013704 000667  BR        3$      ;RETURN
80
81 013706          ERRHRD  17,EM17,ERRO  ;UNRECOVERABLE ERROR ;ERRMSK=17
    013706 104456  TRAP      C$ERHRD
    013710 000021  .WORD    17
    013712 007531  .WORD    EM17
    013714 007722  .WORD    ERRO
82 013716 000662  BR        3$      ;RETURN
83
84 013720          ERRHRD  20,EM20,ERRO  ;RANDOM WRT CHK ERRORS ;ERRMSK=20
    013720 104456  TRAP      C$ERHRD
    013722 000024  .WORD    20
    013724 007576  .WORD    EM20
    013726 007722  .WORD    ERRO
85 013730 000655  BR        3$      ;RETURN
89
90 013732          6$:

```



```

1      .SBTTL  ADJUST DEFECT TO TRACK DESCRIPTOR
2
3      ;*****
4      ;THIS MODULE IS USED TO ADJUST ANY DEFECT DATA FOUND WHICH OCCURS SUBSEQUENT
5      ;TO A TD MOVED AND/OR A HDR 0 MOVED TRACK DESCRIPTOR VALUE.  IT ALSO COUNTS
6      ;THESE DEFECTS IN THE OVERALL COUNTER.
7      ;*
8      ;INPUTS ARE:  BUFFER CALLED 'TDCPY3' (WHICH IS POINTED TO BY R4)
9      ;
10     ;OUTPUTS ARE:  DEFSEC, BUFFER CALLED 'TDCPY3' (WHICH IS POINTED TO BY R4)
11     ;*
12     ;CALL
13     ;      JSR      PC,TDCASE
14     ;*****
15
16 013732 012700 002352      TDCASE:  MOV      #TDLMTS,R0      ;GET POINTER TO TD DEFECT LIMITS
17 013736 012702 000055      MOV      #<TD+G2>,R2      ;INCLUDE THIS IN THE CALCULATION
18 013742 021420              CMP      (R4),(R0)+      ;WAS TD MOVED ?
19 013744 003426              BLE      2$              ;BR IF YES
20 013746 032737 040000 031414  BIT      #BIT14,TDWRD1    ;TD BIT ALREADY SET ?
21 013754 001022              BNE      2$              ;BR IF YES
22 013756 021420              CMP      (R4),(R0)+      ;WAS HDR 0 MOVED ?
23 013760 003407              BLE      1$              ;BR IF YES
24 013762 032737 100000 031414  BIT      #BIT15,TDWRD1    ;HDR 0 BIT ALREADY SET ?
25 013770 001003              BNE      1$              ;BR IF YES
26 013772 012702 000155      MOV      #<G1+TD+G2>,R2  ;INCLUDE THIS IN THE CALCULATION
27 013776 000445              BR       5$
28
29 014000 005337 002410      1$:    DEC      DEFTRK      ;ACCOUNT FOR HEADER 0 MOVED
30 014004 162702 000055      SUB      #<TD+G2>,R2      ;ONE LESS DEFECT TO GO ON THIS TRACK
31 014010 016046 177776      MOV      -2(R0),-(SP)     ;INCLUDE THIS IN THE CALCULATION
32 014014 162416              SUB      (R4),(SP)        ;GET HDR 0 MOVED LIMIT
33 014016 062602              ADD      (SP)+,R2         ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
34 014020 000420              BR       3$              ;AND ADD DIFFERENCE TO THIS CALCULATION.
35
36 014022 005337 002410      2$:    DEC      DEFTRK      ;ACCOUNT FOR TRACK DESCRIPTOR MOVED
37 014026 016446 000002      MOV      2(R4),-(SP)      ;ONE LESS DEFECT TO GO ON THIS TRACK
38 014032 016046 177776      MOV      -2(R0),-(SP)     ;SAVE NEXT DEFECT VALUE
39 014036 162416              SUB      (R4),(SP)        ;GET TD MOVED LIMIT
40 014040 061602              ADD      (SP),R2         ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
41 014042 162616              SUB      (SP)+,(SP)       ;AND ADD DIFFERENCE TO THIS CALCULATION.
42 014044 005720              TST      (R0)+            ;ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
43 014046 022620              CMP      (SP)+,(R0)+      ;UPDATE LIMIT POINTER TO LOOK FOR HDR 0 MOVED
44 014050 003753              BLE      1$              ;WAS HDR 0 ALSO MOVED ?
45 014052 032737 100000 031414  BIT      #BIT15,TDWRD1    ;BR IF YES
46 014060 001347              BNE      1$              ;HDR 0 BIT ALREADY SET ?
47
48 014062 021427 140000      3$:    CMP      (R4),#NULL      ;ANY MORE DEFECTS ?
49 014066 001412              BEQ      6$              ;BR IF NO
50 014070 032737 140000 031414  BIT      #BIT14!BIT15,TDWRD1 ;DID TD AND/OR HDR 0 REALLY GET MOVED ?
51 014076 001003              BNE      4$              ;BR IF YES
52 014100 162702 000066      SUB      #<DS/2>,R2       ;EXCLUDE HALF A SKIP IN THE CALCULATION
53 014104 000402              BR       5$
54 014106 062702 000066      4$:    ADD      #<DS/2>,R2       ;INCLUDE HALF A SKIP IN THE CALCULATION
55 014112 160214      5$:    SUB      R2,(R4)        ;SUBTRACT ALONG WITH SKIP DEFECT INFO
56 014114 000207      6$:    RTS      PC

```

```

1
2
3      ;*****
4      ;THIS MODULE IS USED TO CHECK FOR ANY MINIMUM DATA SEGMENT VIOLATIONS
5      ;PRODUCED BY 'CALCTD'.
6      ;*
7      ;INPUTS ARE:   FROM (R5), WHICH IS A BUFFER CALLED 'TDSBLK', FROM (R1),
8      ;              WHICH IS A BUFFER CALLED 'DELTA'.  CONSTANTS USED ARE THOSE
9      ;              WHICH DETERMINE A LEADING OR TRAILING DATA SEGMENT DEFECT
10     ;              SKIP VIOLATION.
11     ;
12     ;OUTPUTS ARE:  TO (R5), WHICH IS A BUFFER CALLED 'TDSBLK', TO (R1), WHICH
13     ;              IS A BUFFER CALLED 'DELTA', TO (R4), WHICH IS A BUFFER
14     ;              CALLED 'TDCPY3'.
15     ;*
16     ;THIS MODULE IS CALLED BY: CALCTD
17     ;*****
18 014116 010537 002250  CHKSEG: MOV     R5,TEMPA      ;SAVE POINTER TO THIS DEFECT
19 014122 010046          MOV     R0,-(SP)        ;PUSH R0 ON STACK
20 014124 010246          MOV     R2,-(SP)        ;PUSH R2 ON STACK
21 014126 005000          CLR     R0          ;RESET R0 (SEGMENT LEADING VS TRAILING)
22 014130 012701 002570  MOV     @DELTA,R1      ;GET POINTER TO BEGINNING TO DELTA TABLE
23 014134 013746 002406  MOV     DEFSEC,-(SP)   ;GET THE # OF DEFECTS IN THIS SECTOR
24 014140 005745          1$:  TST     -(R5)        ;BACK UP THE DEFECT POINTER FOR THIS SECTOR
25 014142 005316          DEC     (SP)        ;DONE ALL DEFECTS FOR THIS SECTOR YET ?
26 014144 003375          BGT    1$          ;BR IF NO
27 014146 005726          TST     (SP)+       ;RESTORE THE STACK
28 014150 012737 000042 002412  MOV     @34.,MINSEG    ;SETUP THE LEADING MINIMUM DATA SEGMENT VALUES
29 014156 012737 000064 002414  MOV     @52.,MAXSEG    ;MIN. PARTITION FOR DATA SEGMENT
30 014164 011502          MOV     (R5),R2      ;MAX. PARTITION FOR DATA SEGMENT
31                                     ;GET BEGINNING OF DEFECT
32                                     ;NOW LOOK TO SEE IF THE BEGINNING OF THE DEFECT
33                                     ;FALLS WITHIN GAP 3, BUT DON'T ALLOW THE DEFECT
34 014166 020237 002412          CMP     R2,MINSEG     ;SKIP TO START ON THE LAST WORD OF GAP 3
35 014172 103425          BLO    5$          ;IS BEGINNING OF DEFECT IN GAP 3 ?
36 014174 000404          BR     4$          ;BR IF YES
37 014176 061502          2$:  ADD     (R5),R2      ;ADD BEGINNING OF DEFECT TO CALCULATION
38                                     ;NOW LOOK TO SEE IF THE BEGINNING OF THE
39                                     ;DEFECT FALLS WITHIN THE MINIMUM DATA SEGMENT
40 014200 020237 002412          3$:  CMP     R2,MINSEG     ;IS BEGINNING OF DEFECT IN MINIMUM DATA SEGMENT ?
41 014204 101420          BLOS   5$          ;BR IF NO
42 014206 020237 002414          4$:  CMP     R2,MAXSEG     ;IS BEGINNING OF DEFECT IN MINIMUM DATA SEGMENT ?
43 014212 101030          BHI    6$          ;BR IF NO
44 014214 005711          TST     (R1)        ;WAS THIS DEFECT PREVIOUSLY ADJUSTED ?
45 014216 001013          BNE    5$          ;BR IF YES
46 014220 013746 002414          MOV     MAXSEG,-(SP)  ;PUSH MAXSEG ON STACK
47 014224 160216          SUB     R2,(SP)      ;FIND THE SKEW VALUE AND
48 014226 061615          ADD     (SP),(R5)    ;CORRECT THIS DEFECT VALUE
49 014230 012611          MOV     (SP)+,(R1)   ;SAVE THE VALUE IN THE DELTA TABLE
50 014232 026427 000002 140000  CMP     2(R4),@NULL   ;ANY MORE DEFECTS ?
51 014240 001402          BEQ    5$          ;BR IF NO
52 014242 161164 000002          SUB     (R1),2(R4)   ;REMOVE THE DELTA VALUE FROM NEXT DEFECT
53
54 014246 062737 000154 002412  5$:  ADD     @<DS>,MINSEG   ;ADD A DEFECT SKIP TO THE MIN. SEGMENT VALUE
55 014254 062737 000154 002414  ADD     @<DS>,MAXSEG   ;AND A DEFECT SKIP TO THE MAX. SEGMENT VALUE
56 014262 022521          CMP     (R5)+,(R1)+  ;POP THE DATA AND DELTA FILES
    
```

57	014264	020537	002250		CMP	R5,TEMPA		;DONE CHECKING DATA SEGMENTS YET ?
58	014270	001413			BEQ	7:		;BR IF YES
59	014272	000741			BR	2:		;ELSE KEEP GOING...
60								
61	014274	005700		6:	TST	R0		;HAVE WE LOOKED AT THE TRAILING DATA SEGMENT YET ?
62	014276	100763			BMI	5:		;BR IF YES
63								;SETUP THE TRAILING MINIMUM DATA SEGMENT VALUES
64	014300	062737	000360	002412	ADD	#240.,MINSEG		;MIN. PARTITION FOR DATA SEGMENT
65	014306	062737	000360	002414	ADD	#240.,MAXSEG		;MAX. PARTITION FOR DATA SEGMENT
66	014314	005100			COM	R0		;MARK THIS EVENT
67	014316	000730			BR	3:		;TRY AGAIN...
68	014320			7:				
	014320	012602			MOV	(SP)+,R2		;POP STACK INTO R2
	014322	012600			MOV	(SP)+,R0		;POP STACK INTO R0
69	014324	000207			RTS	PC		;TAKE THE RETURN

```

1      .SBTTL TRACK DESCRIPTOR CALCULATION MODULE
2
3      ;*****
4      ;THIS ROUTINE COMPUTES THE ACTUAL OFFSET OF THE TD WORD, AND DETERMINES IF THE
5      ;DEFECT LIES WITHIN THE CURRENT SECTOR. IF THE DEFECT DOES, THE APPROPRIATE
6      ;CALCULATION IS MAPPED AND PREPARED FOR TRANSFER TO THE DRIVE.
7      ;*
8      ;INPUTS ARE: TDWRD1 - TDWRD6, DS, PARTBL, SIZE OF SECTOR (HDR+G3+DATA+G4)
9      ;
10     ;OUTPUTS ARE: TDSBLK, R0 - R5, DEFSEC, DEFTRK, TDCPY3 - TDCPY6,
11     ;DEF1 - DEF4, DEFCNT
12     ;*
13     ;THIS MODULE IS CALLED BY: MAKTRK
14     ;*****
15
16 014326 005037 002410          CALCTD: CLR    DEFTRK      ;RESET THE # OF DEFECTS PER TRACK COUNTER
17 014332 012705 033614          MOV    #TDSBLK,R5    ;POINT TO THE TD OUTPUT BLOCK
18 014336 013737 002230 002232  MOV    ENDPTR,ENDTAB ;GET THE BUFFER SIZE LIMIT
19 014344 006337 002232          ASL    ENDTAB        ;PUT THE LIMIT ON A WORD BOUNDARY
20 014350 062737 033614 002232  ADD    #TDSBLK,ENDTAB ;AND SET THE LIMIT
21 014356 012704 031430          1#:  MOV    #TDCPY1,R4   ;RESTORE POINTER TO 1ST TD WORD
22 014362 013724 031414          MOV    TDWRD1,(R4)+  ;SAVE TD WORD #1
23 014366 013724 031416          MOV    TDWRD2,(R4)+  ;SAVE TD WORD #2
24 014372 013724 031420          MOV    TDWRD3,(R4)+  ;SAVE TD WORD #3
25 014376 013724 031422          MOV    TDWRD4,(R4)+  ;SAVE TD WORD #4
26 014402 013724 031424          MOV    TDWRD5,(R4)+  ;SAVE TD WORD #5
27 014406 013724 031426          MOV    TDWRD6,(R4)+  ;SAVE TD WORD #6
28 014412 012704 031434          MOV    #TDCPY3,R4   ;RESTORE POINTER TO 3RD TD WORD (1ST DEFECT)
29 014416 022427 140000          CMP    (R4)+,#NULL  ;ANY DEFECTS ON THIS TRACK ?
30 014422 001007                  BNE    3#           ;BR IF YES
31 014424                  2#:  ;NO MORE DEFECTS ON THIS TRACK, SO NULL THE
32                                ;SKIP DISPLACEMENT WORDS IN EACH OF THE
33                                ;REMAINING SECTORS.
33 014424 012725 140000          MOV    #NULL,(R5)+  ;DONE WITH THE TABLE YET ?
34 014430 020537 002232          CMP    R5,ENDTAB   ;BR IF NO
35 014434 003773                  BLE    2#           ;EXIT, GO WRITE FORMAT FOR THIS TRACK
36 014436 000137 015166          JMP    17#         ;NOW CORRECT EACH SKIP TO THE BEGINNING OF
37                                ;THE DEFECT, SO IT CAN BE INSERTED INTO IT'S
38                                ;APPROPRIATE SECTOR.
39                                ;ACCUMULATE # OF DEFECTS ON THIS TRACK
40 014442 005237 002410          3#:  INC    DEFTRK      ;DID WE REACH LAST DEFECT ?
41 014446 023727 002410 000004  CMP    DEFTRK,#4   ;BR IF YES
42 014454 103003                  BHS    4#           ;ANY MORE DEFECTS ?
43 014456 022427 140000          CMP    (R4)+,#NULL  ;BR IF YES
44 014462 001367                  BNE    3#           ;NOW GO LOOK TO SEE IF THE TD AND/OR HDR 0 HAVE
45                                ;BEEN MOVED, SO THE REMAINING DEFECTS CAN BE
46                                ;ADJUSTED ACCORDINGLY.
47                                ;RESTORE POINTER TO 3RD TD WORD (1ST DEFECT)
48 014464 012704 031434          4#:  MOV    #TDCPY3,R4   ;LOOK FOR A TD AND/OR HDR 0 MOVED
49 014470 004737 013732          JSR    PC,TDCASE    ;ANY MORE DEFECTS ?
50 014474 021427 140000          CMP    (R4)+,#NULL  ;BR IF NO
51 014500 001751                  BEQ    2#           ;NOW FIND WHICH SECTOR THE DEFECT SHOULD BE
52                                ;INSERTED INTO.
53                                ;CORRECT 1ST DEFECT TO THE BEGINNING OF DEFECT
54 014502 162714 000066          5#:  SUB    #<DS/2>,(R4) ;RESET THE SAME SECTOR FLAG,
55 014506 105037 002245          CLRB   SAMSEC       ;THE # OF DEFECTS PER SECTOR COUNTER AND
56 014512 005037 002406          CLR    DEFSEC       ;THE TEMP. ACCUMULATIVE DELTA COUNTER.
57 014516 005037 015170          CLR    TMDLTA
    
```

58	014522	012700	000004		MOV	#4,R0	;GET # OF ENTRIES
59	014526	012701	002570		MOV	#DELTA,R1	;GET POINTER TO BEGINNING OF DELTA TABLE
60	014532	005021		6#:	CLR	(R1)+	;INITIALIZE THE TABLE
61	014534	005300			DEC	R0	;DONE YET ?
62	014536	003375			BGT	6#	;BR IF NO
63	014540	012701	002570		MOV	#DELTA,R1	;GET POINTER TO BEGINNING OF DELTA TABLE
64	014544	005714			TST	(R4)	;ANY MORE DEFECTS ON TRACK ?
65	014546	003726			BLE	2#	;BR IF NO
66	014550	012700	000517		MOV	#<HDR+G3+DATA+G4>,R0	;GET THE SIZE OF SECTOR
67	014554	160014			SUB	R0,(R4)	;IS THE DEFECT IN THIS SECTOR ?
68	014556	003166			BGT	16#	;BR IF NO
69	014560	060014			ADD	R0,(R4)	;YES, RESTORE DEFECT
70							;NOW FIND WHERE THE BEGINNING OF THE DEFECT
71							;FALLS WITHIN THIS PARTICULAR SECTOR
72	014562	012703	002336		MOV	#PARTBL,R3	;GET POINTER TO BEGINNING OF SECTOR PARTITION TABLE
73	014566	012302			MOV	(R3)+,R2	;GET THE 1ST PARTITION
74	014570	021402		7#:	CMP	(R4),R2	;IS THE DEFECT IN THIS PART OF THE SECTOR ?
75	014572	101410			BLOS	9#	;BR IF YES
76	014574	021400			CMP	(R4),R0	;IS THIS DEFECT AT THE SECTOR LIMIT ?
77	014576	001402			BEQ	8#	;BR IF YES
78	014600	062302			ADD	(R3)+,R2	;EXTEND THE PARTITION RANGE IN THE SECTOR
79	014602	000772			BR	7#	;AND TRY AGAIN
80	014604	005314		8#:	DEC	(R4)	;FOUND DEFECT AT END OF SECTOR, REDUCE BY ONE
81	014606	005237	002406		INC	DEFSEC	;ONE MORE DEFECT FOUND IN THIS SECTOR
82	014612	000425			BR	10#	;TO FACILLITATE RP07 MICROCODE TESTING
83							
84	014614	005237	002406	9#:	INC	DEFSEC	;ONE MORE DEFECT FOUND IN THIS SECTOR
85	014620	105737	002245		TSTB	SAMSEC	;IS THIS THE 1ST DEFECT IN THIS SECTOR ?
86	014624	001420			BEQ	10#	;BR IF YES
87							;NOW SEE IF THE BEGINNING OF THIS DEFECT FALLS
88							;WITHIN THE NEXT MINIMUM DATA SEGMENT OF THIS
89							;SECTOR
90	014626	021427	000176		CMP	(R4),#<DS+MSEG>	;IS DEFECT WITHIN NEXT MINIMUM DATA SEGMENT ?
91	014632	101015			BHI	10#	;BR IF NO
92	014634	012746	000176		MOV	#<DS+MSEG>,-(SP)	;PUSH #<DS+MSEG> ON STACK
93	014640	011615			MOV	(SP),(R5)	;SAVE THIS VALUE IN THE DEFECT TABLE
94	014642	161416			SUB	(R4),(SP)	;GET DELTA FOR SKEW FORWARD TO THE MINIMUM DATA SEGMENT
95	014644	012621			MOV	(SP)+,(R1)+	;SAVE THE VALUE IN THE DELTA TABLE
96	014646	026427	000002	140000	CMP	2(R4),#NULL	;ANY MORE DEFECT ?
97	014654	001431			BEQ	12#	;BR IF NO
98	014656	166164	177776	000002	SUB	-2(R1),2(R4)	;ADJUST THE NEXT DEFECT BY DELTA
99	014664	000425			BR	12#	;NOW, GO AND SET THE STATUS
100							
101	014666	011415		10#:	MOV	(R4),(R5)	;LOG THIS NON-CRITICAL INFORMATION
102	014670	105737	002245		TSTB	SAMSEC	;IS THIS THE 1ST DEFECT IN THIS SECTOR ?
103	014674	001021			BNE	12#	;BR IF NO
104							;NOW SEE IF THE BEGINNING OF THIS DEFECT FALLS
105							;WITHIN THE HEADER OF THIS SECTOR
106	014676	021527	000010		CMP	(R5),#<HDR>	;IS THE DEFECT WITHIN THE HEADER ?
107	014702	101014			BHI	11#	;BR IF NO
108	014704	012746	000010		MOV	#<HDR>,-(SP)	;PUSH #<HDR> ON STACK
109	014710	161516			SUB	(R5),(SP)	;FIND THE SKEW VALUE AND
110	014712	061615			ADD	(SP),(R5)	;CORRECT THIS DEFECT VALUE
111	014714	012621			MOV	(SP)+,(R1)+	;SAVE THE VALUE IN THE DELTA TABLE
112	014716	026427	000002	140000	CMP	2(R4),#NULL	;ANY MORE DEFECT ?
113	014724	001403			BEQ	11#	;BR IF NO
114	014726	166164	177776	000002	SUB	-2(R1),2(R4)	;ADJUST THE NEXT DEFECT BY DELTA

115	014734	162715	000006	11#:	SUB	#<HDR-2>,(R5)	;NOW MAKE THE REFERENCE TO THE BEGINNING
116							;OF THE 1ST HEADER CRC WORD
117							;AND POINT TO THE NEXT DEFECT ENTRY
118	014740	005725		12#:	TST	(R5)+	;MOVE THE OUTPUT POINTER UP ONCE!
119	014742	004737	014116		JSR	PC,CHKSEG	;GO CHECK FOR MINIMUM DATA SEGMENT VIOLATIONS
120	014746	005337	002410		DEC	DEFTRK	;ONE LESS DEFECT TO GO ON THIS TRACK
121	014752	001624			BEQ	2#	;IF ZERO, TRACK RECORD IS COMPLETE
122	014754	013746	015170		MOV	TMDLTA,-(SP)	;PUSH TMDLTA ON STACK
123	014760	163716	002570		SUB	DELTA,(SP)	;ACCUMULATE ALL DELTA VALUES
124	014764	163716	002572		SUB	DELTA*2,(SP)	;SO THEY CAN BE SUBTRACTED FROM
125	014770	163716	002574		SUB	DELTA*4,(SP)	;THE BALANCE OF THE SECTOR REMAINS.
126	014774	163726	002576		SUB	DELTA*6,(SP)+	;ANY MORE DELTA'S ACCUMULATED ?
127	015000	002005			BGE	13#	;BR IF NO
128	015002	005446			NEG	-(SP)	;MAKE THE DIFFERENCE POSITIVE
129	015004	061637	015170		ADD	(SP),TMDLTA	;AND ADD IT TO OLD ACCUMULATED TEMP. DELTA
130	015010	161600			SUB	(SP),R0	;SUBTRACT DELTA FROM END OF SECTOR BOUNDARY AND
131	015012	162602			SUB	(SP)+,R2	;FROM SECTOR PARTITION VALUE.
132	015014	161400		13#:	SUB	(R4),R0	;MAKE THE END OF SECTOR BOUNDARY RELATIVE TO
133	015016	062700	000154		ADD	#<DS>,R0	;THE PREVIOUS DEFECT AND ADD A SKIP TO EXTEND
134							;THE END OF SECTOR BOUNDARY.
135	015022	162402			SUB	(R4)+,R2	;MAKE THE SECTOR PARTITIONS RELATIVE TO THE
136	015024	062702	000154		ADD	#<DS>,R2	;PREVIOUS DEFECT AND ADD A SKIP TO EXTEND
137							;THE PARTITION RANGE.
138	015030	021427	000154		CMP	(R4),#<DS>	;WILL THIS DEFECT BE STACKED ?
139	015034	003406			BLE	14#	;BR IF YES
140	015036	021400			CMP	(R4),R0	;IS THE BEGINNING OF DEFECT IN CURRENT SECTOR ?
141	015040	003023			BGT	15#	;BR IF NO
142	015042	001660			BEQ	8#	;BR IF AT THE END OF THE CURRENT SECTOR
143	015044	105237	002245		INCB	SAMSEC	;MARK THIS DEFECT AS BEING IN SAME SECTOR
144	015050	000647			BR	7#	
145							;NOW SETUP TO ADJUST THE SECTOR FOR STACKED
146							;DEFECTS
147	015052	005237	002406	14#:	INC	DEFSEC	;ONE MORE DEFECT FOUND IN THIS SECTOR
148	015056	012715	000154		MOV	#<DS>,(R5)	;LOG THE STACKED DEFECT INFORMATION
149	015062	026427	000002	140000	CMP	2(R4),#NULL	;ANY MORE DEFECTS ?
150	015070	001723			BEQ	12#	;BR IF NO
151	015072	012746	000154		MOV	#<DS>,-(SP)	;VALUE OF STACKED SKIP
152	015076	161416			SUB	(R4),(SP)	;SUBTRACT CURRENT VALUE OF R4 FROM STACKED
153	015100	161664	000002		SUB	(SP),2(R4)	;VALUE AND ADJUST NEXT DEFECT BY DELTA
154	015104	012621			MOV	(SP)+,(R1)+	;LOG THE DELTA
155	015106	000714			BR	12#	
156							
157	015110	160014		15#:	SUB	R0,(R4)	;MODIFY THE SECTOR REMAINDER
158	015112	006337	002406		ASL	DEFSEC	;ADJUST DEFECT COUNT X 2
159	015116	006337	002406		ASL	DEFSEC	;ADJUST DEFECT COUNT X 4
160	015122	062737	015134	002406	ADD	#16#,DEFSEC	;NOW FIND HOW MANY DEFECTS TO NULL OUT IN THE
161	015130	000177	165252		JMP	#DEFSEC	;REMAINDER OF THE HEADER WORDS IN THIS SECTOR
162							
163	015134	012725	140000	16#:	MOV	#NULL,(R5)+	;NULL SD1,
164	015140	012725	140000		MOV	#NULL,(R5)+	;NULL SD2,
165	015144	012725	140000		MOV	#NULL,(R5)+	;NULL SD3,
166	015150	012725	140000		MOV	#NULL,(R5)+	;NULL SD4 IN THE HEADER OF THIS SECTOR
167	015154	020537	002232		CMP	R5,ENDTAB	;DID WE DO THE ENTIRE TRACK YET ?
168	015160	101002			BHI	17#	;BR IF YES
169	015162	000137	014506		JMP	5#	;AND CONTINUE
170	015166	000207		17#:	RTS	PC	;EXIT, GO WRITE FORMAT FOR THIS TRACK
171							

F6

172 015170 000000

TMDLTA: .WORD 0

;TEMP. ACCUMULATIVE DELTA VALUE GOES HERE

```

1          .SBTTL  CHECK BAD SECTOR FILE ENTRIES
2
3          ;;*****
4          ;THIS MODULE CHECKS TO SEE IF THE CYLINDER ADDRESS BEING SETUP BY THE 'MAKTRK'
5          ;ROUTINE IS ENTERED IN MFG BAD SECTOR FILE.
6          ;*
7          ;INPUTS ARE:   TMPBSF, TMPBSF+2
8          ;
9          ;OUTPUTS ARE:
10         ;*
11         ;CALL
12         ;       JSR      PC,CHKBSF      ;NOW CHECK THE BAD SECTOR FILE
13         ;       ----- ;RETURN HERE IF SECTOR NOT FOUND,
14         ;       ----- ;ELSE RETURN HERE
15         ;*
16         ;THIS MODULE IS CALLED BY: MERGE
17         ;;*****
18
19         CHKBSF:
20         MOV      RO,-(SP)      ;;PUSH RO ON STACK
21         MOV      #MFGBUF+6.,RO ;GET OVER SERIAL NUMBER IN BSF TABLE
22         1$:      TST      (RO)+  ;UPDATE POINTER TO NEXT CYLINDER BOUNDARY
23         ;         ;IN THE BSF TABLE.
24         2$:      TST      (RO)   ;IS THIS THE TERMINATOR OF THE BSF ?
25         BMI      4$          ;BR IF YES (SECTOR NOT FOUND)
26         CMP      RO,#ENDBSF  ;IS THIS THE END OF THE BSF ENTRIES ?
27         BHI      4$          ;BR IF YES (SECTOR NOT FOUND)
28         CMP      TMPBSF,(RO)+ ;SAME CYLINDER ADDRESS ?
29         BNE      1$          ;BR IF NO
30         CMP      TMPBSF+2,(RO)+ ;SAME TRK/SEC ADDRESS ?
31         BNE      2$          ;BR IF NO
32         3$:      ADD      @2,2(SP) ;ADJUST RETURN FOR SECTOR FOUND
33         4$:
34         MOV      (SP)+,RO      ;;POP STACK INTO RO
          RTS      PC

```



```

1      .SBTTL FILE MERGE MODULE FOR TRACK RECORD
2
3      ;*****
4      ;THIS MODULE CREATES THE TRACK RECORD WHICH WILL BECOME THE FORMAT ON AN RP07,
5      ;FOR A GIVEN TRACK.
6      ;*
7      ;INPUTS ARE: INTLEV, PLTRK, TKWCNT, DESCYL, DESTRK, TDSBLK
8      ;
9      ;OUTPUTS ARE: HDRBLK*0 - HDRBLK*300
10     ;*
11     ;THIS MODULE IS CALLED BY: MAKTRK
12     ;*****
13
14 015242 005001          MERGE:: CLR      R1          ;SET THE TOGGLE SWITCH TO NON-INTERLEAVED
15 015244 012705 033614  MOV      #TDSBLK,R5 ;GET THE INITIAL TD TABLE ADDRESS,
16 015250 012700 002323  MOV      #PLTRK,R0   ; THE RP07 INTERLEAVE MASK,
17 015254 112003          MOV      (R0)+,R3   ; THE 1ST INTERLEAVED SECTOR ADDRESS,
18 015256 111004          MOV      (R0),R4    ; THE 1ST SECTOR ADDRESS,
19 015260 012700 032464  MOV      #HDRBLK,R0   ; THE HEADER TABLE ADDRESS AND
20 015264 013737 002364 002232  MOV      TKWCNT,ENDTAB ; THE # OF WORDS PER TRACK (6 WRDS X 50. SECTORS).
21 015272 006337 002232  ASL      ENDTAB   ;ADJUST WORDS TO BYTES (X 2) AND
22 015276 062737 032464 002232  ADD      #HDRBLK,ENDTAB ;CALULATE END OF HEADER TABLE.
23 015304 012705 033614  MOV      #TDSBLK,R5   ;GET POINTER TO BEGINNING OF TD BLOCK
24 015310 053704 002314 1# : BIS      DESTRK,R4   ;ADD THE TRACK ADDRESS TO
25 015314 053703 002314  BIS      DESTRK,R3   ;BOTH ADDRESS COUNTERS,
26 015320 013720 002312 2# : MOV      DESCYL,(R0)+ ;LOAD THE CYLINDER ADDRESS
27 015324 005701          TST      R1          ;SEE IF INTERLEAVED SECTOR SHOULD BE LOADED,
28 015326 001404          BEQ      3#          ;BRANCH IF NOT, ELSE
29 015330 010310          MOV      R3,(R0)    ;LOAD INTERLEAVED SECTOR,
30 015332 005203          INC      R3          ;INCREMENT TO NEXT SECTOR ADDRESS,
31 015334 005001          CLR      R1          ;SET THE TOGGLE SWITCH TO NON-INTERLEAVED AND
32 015336 000406          BR       4#          ;CONTINUE.
33
34 015340 010410          3# : MOV      R4,(R0)    ;LOAD LOW OR NON-INTERLEAVED SECTOR,
35 015342 005204          INC      R4          ;INCREMENT TO NEXT SECTOR ADDRESS.
36 015344 105737 002244  TSTB    INTLEV      ;IS THE INTERLEAVE SWITCH ON?
37 015350 001401          BEQ      4#          ;BR IF NO
38 015352 005101          COM      R1          ;SET THE TOGGLE SWITCH TO INTERLEAVED
39 015354 011037 031462 4# : MOV      (R0),TMPBSF+2 ;SETUP THE TRACK/SECTOR,
40 015360 014037 031460  MOV      -(R0),TMPBSF ;CYLINDER TO BE CHECKED FOR IN THE BSF.
41 015364 004737 015172  JSR     PC,CHKBSF   ;NOW CHECK THE BAD SECTOR FILE,
42 015370 000403          BR       5#          ;RETURN HERE IF SECTOR NOT FOUND,
43                                     ;ELSE RETURN HERE.
44 015372 052720 050000  BIS     #050000,(R0)+ ;SET UF/FMT BITS IN CYLINDER ADDRESS
45 015376 000402          BR       6#          ;
46
47 015400 052720 150000 5# : BIS     #150000,(R0)+ ;SET MF/UF/FMT BITS IN CYLINDER ADDRESS
48 015404 005720 6# : TST     (R0)+      ;GET POINTER OVER TRK/SEC ENTRY
49 015406 012520          MOV     (R5)+,(R0)+ ;LOG THE TD DATA
50 015410 012520          MOV     (R5)+,(R0)+ ;ALL FOUR WORDS OF IT
51 015412 012520          MOV     (R5)+,(R0)+ ;FOR EACH SECTOR
52 015414 012520          MOV     (R5)+,(R0)+ ;THIS IS LAST WORD FOR EACH SECTOR
53 015416 023700 002232  CMP     ENDTAB,R0   ;IS THIS THE END OF THE HEADER TABLE ?
54 015422 101412          BLOS   8#          ;BR F YES
55 015424 123704 002322  CMPB   SEC50,R4    ;DONE WITH ALL SECTORS ON THIS TRACK ?
56 015430 103002          BHIS  7#          ;BR IF NO
57 015432 005004          CLR     R4          ;RESET R4 TO SECTOR ADDRESS 0 AND
    
```

58	015434	000725		BR	1#		;CONTINUE.
59							
60	015436	123703	002322	7#:	CMPB	SEC50,R3	;DONE WITH ALL SECTORS ON THIS TRACK ?
61	015442	103326			BHIS	2#	;BR IF NO
62	015444	005003			CLR	R3	;RESET R3 TO SECTOR ADDRESS 0 AND
63	015446	000720			BR	1#	;CONTINUE.
64							
65	015450	000207		8#:	RTS	PC	;RETURN

```

1      .SBTTL MAKE TRACK BUFFER
2
3      ;*****
4      ;THIS MODULE IS USED ANYTIME A FORMAT TRACK OR VERIFY TRACK COMMAND IS ISSUED.
5      ;THE MODULE CREATES THE BUFFER USED FOR THE OPERATION AND DOES THE SETUP FOR
6      ;THE COMMAND EXECUTION. IT ALSO EXECUTES THE COMMAND AND ALLOWS A RETRY IF
7      ;THE COMMAND FAILS.
8      ;*
9      ;INPUTS ARE:  FUNCTN, TKWCNT
10     ;
11     ;OUTPUTS ARE:  ERRFNC, BUSADR, NEGWRD, RPOF
12     ;*
13     ;CALL
14     ;      JSR      PC,MAKTRK      ;NOW EXECUTE THE COMMAND
15     ;      -----      ;RETURN HERE IF EXCEEDED RETRY LIMIT
16     ;      -----      ;ELSE RETURN HERE
17     ;*
18     ;THIS MODULE IS CALLED BY: MAIN
19     ;*****
20
21     MAKTRK:
22     015452 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
23     015454 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
24     015456 010246      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
25     015460 010346      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
26     015462 010446      MOV      R4,-(SP)      ;;PUSH R4 ON STACK
27     015464 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
28     015466 004737 014326  JSR      PC,CALCTD     ;DO THE DEFECT LOCATING NOW
29     015472 004737 015242  JSR      PC,MERGE      ;NOW PUT THE DEFECTS IN THE CORRECT SECTOR
30     015476 012605      MOV      (SP)+,R5      ;;POP STACK INTO R5
31     015500 012604      MOV      (SP)+,R4      ;;POP STACK INTO R4
32     015502 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
33     015504 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
34     015506 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
35     015510 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
36     015512 022737 000063 002404 1$:  CMP      #FMTRK,FUNCTN ;IS THIS A FORMAT COMMAND?
37     015520 001004      BNE      2$            ;BR IF NO
38     015522 012737 005700 002260      MOV      #FORMAT,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
39     015530 000413      BR       3$            ;AND SKIP NEXT
40
41     015532 022737 000073 002404 2$:  CMP      #RDHD,FUNCTN  ;IS THIS A VERIFY FUNCTION?
42     015540 001007      BNE      3$            ;BR IF NO (MUST BE A WRITE CHECK)
43     015542 012737 005713 002260      MOV      #RMDATA,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
44     015550 012737 033614 002234      MOV      #TDSBLK,BUSADR ;GET POINTER TO BEGINNING OF TD BLOCK
45     015556 000403      BR       4$            ;AND GO-ON
46
47     015560 012737 032464 002234 3$:  MOV      #HDRBLK,BUSADR ;GET POINTER TO BEGINNING OF HEADER BLOCK
48     015566 013737 002364 002376 4$:  MOV      TKWCNT,NEGWRD  ;SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
49     015574 052737 100000 002236      BIS      #CMOD,OFFSET  ;SET COMMAND MODIFIER BIT
50     015602 004737 010502      JSR      PC,DRIVER     ;EXECUTE THE OPERATION NOW
51     015606 004737 013304      JSR      PC,RETRY      ;SEE IF WE HAD ANY ERRORS
52     015612 000405      BR       5$            ;RETURN HERE IF EXCEEDED RETRY LIMIT
53
54     015614 005737 002274      TST      RTYCNT        ;DID WE HAVE AN ERROR ?
55     015620 001334      BNE      1$            ;BR IF YES
56     015622 062716 000002      ADD      #2,(SP)       ;ADJUST FOR GOOD RETURN
57     015626 000207      RTS      PC            ;TAKE RETURN NOW
    
```

```

1
2
3
4
5
6
7
8
9
10
11 015630 004737 013256      DRVINI: JSR      PC,RESET      ;SEIZE THE PORT AND RESET THE MASSBUS
12                                     ;PRINT 'DRIVE X'
13 015634 013746 002434      MOV      DRVNO,-(SP)
14 015640 012746 003421      MOV      #FRMT10,-(SP)
15 015644 012746 000002      MOV      #2,-(SP)
16 015650 010600              MOV      SP,RO
17 015652 104417              TRAP     C#PNTF
18 015654 062706 000006      ADD      #6,SP
19 015660 005777 164566      TST      @RPDS              ;DO A READ OF THE DRIVE STATUS REG
20 015664 032777 010000 164556 BIT      @NED,@RPCS2        ;DO WE HAVE NED?
21 015672 001405              BEQ      1$                 ;IF ZERO, DRIVE DOES EXIST
22 015674 104455              TRAP     C#ERDF            ;REPORT 'NON-EXISTENT DRIVE'
23 015676 000001              .WORD   1
24 015700 007654              .WORD   EM21
25 015702 000000              .WORD   0
26 015704 000572              BR       NORUN             ;TAKE THE BAD BRANCH, THE DRIVE IS NO GOOD!
27
28 015706 032777 000004 164536 1$: BIT      @ILEV,@RPDS        ;IS THIS DRIVE INTERLEAVE ENABLED?
29 015714 001414              BEQ      2$                 ;IF BIT = 0, NO
30 015716 112737 000377 002244 MOVB     #377,INTLEV        ;SET THE INTERLEAVE ENABLED MASK
31 015724 012746 004231              MOV      #FRMT31,-(SP)
32 015730 012746 000001              MOV      #1,-(SP)
33 015734 010600              MOV      SP,RO
34 015736 104417              TRAP     C#PNTF
35 015740 062706 000004      ADD      #4,SP
36 015744 000410              BR       3$
37 015746              ;PRINT ', NOT INTERLEAVED'
38 015746 012746 004203      MOV      #FRMT30,-(SP)
39 015752 012746 000001              MOV      #1,-(SP)
40 015756 010600              MOV      SP,RO
41 015760 104417              TRAP     C#PNTF
42 015762 062706 000004      ADD      #4,SP
43 015766 017746 164474      MOV      @RPDT,-(SP)        ;GET THE DRIVE TYPE REGISTER
44 015772 042716 004000      BIC      @BIT11,(SP)        ;CLEAR THE DRIVE REQUEST REQUIRED BIT
45 015776 022726 020042      CMP      #20042,(SP)+      ;IS THE DRIVE AN RP07?
46 016002 001405              BEQ      4$                 ;BR IF YES
47 016004 104455              TRAP     C#ERDF            ;REPORT 'DRIVE NOT AN RP07'
48 016006 000002              .WORD   2
49 016010 007677              .WORD   EM22
50 016012 000000              .WORD   0
51 016014 000526              BR       NORUN
52
53 016016 012701 000004      4$: MOV      #4,R1           ;GET AN OVERALL ITERATION COUNT
54 016022 105777 164424      5$: TSTB     @RPDS          ;IS DRIVE READY SET ?

```

```

39 016026 100415          BMI      6#           ;BR IF YES
40 016030 004737 013106   JSR      PC,WAIT      ;WAIT A LITTLE LONGER
41 016034 005301          DEC      R1           ;DONE TRYING YET ?
42 016036 003371          BGT      5#           ;BR IF NO
43 016040 012737 000003 002272  MOV      #3,ERRMSK    ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
44 016046 012737 005757 002260  MOV      #DRVRDY,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
45 016054 004737 013402   JSR      PC,ERRORS    ;REPORT THE ERROR
46 016060 000504          BR       NORUN        ;DRIVE IS NO GOOD
47
48 016062 032777 010000 164362 6# :   BIT      #MOL,SRPDS   ;IS THE DRIVE ONLINE?
49 016070 001014          BNE      7#           ;BR IF YES
50 016072 012737 005727 002260  MOV      #MOLINE,ERRFNC ;MARK THE CURRENT FUNCTION
51 016100 012737 000005 002272  MOV      #5,ERRMSK    ;LOG 'DRIVE OFFLINE ERROR' STATUS
52 016106 004737 013304   JSR      PC,RETRY     ;SEE IF WE HAD ANY ERRORS
53 016112 000471          BR       NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
54
55 016114 005737 002274          TST      RTYCNT       ;ELSE RETURN HERE
56 016120 001360          BNE      6#           ;DID WE HAVE AN ERROR ?
57
58 016122 005737 002164          TST      VRIFY        ;IS THIS VERIFY ONLY MODE ?
59 016126 001023          BNE      9#           ;BR IF YES
60 016130 005737 002172          TST      LIST         ;IS THIS LIST MODE ?
61 016134 001020          BNE      9#           ;BR IF YES
62 016136 032777 004000 164306 8# :   BIT      #WRL,SRPDS   ;IS THE DRIVE WRITE LOCKED ?
63 016144 001414          BEQ      9#           ;BR IF NO
64 016146 012737 005743 002260  MOV      #WLOCK,ERRFNC ;MARK THE WRITE LOCK TEST
65 016154 012737 000004 002272  MOV      #4,ERRMSK    ;LOG 'DRIVE WRITE LOCKED ERROR' STATUS
66 016162 004737 013304   JSR      PC,RETRY     ;SEE IF WE HAD ANY ERRORS
67 016166 000443          BR       NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
68
69 016170 005737 002274          TST      RTYCNT       ;ELSE RETURN HERE
70 016174 001360          BNE      8#           ;DID WE HAVE AN ERROR ?
71
72 016176 012737 005672 002260 9# :   MOV      #RECALI,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
73 016204 012777 000007 164226  MOV      #RECAL,SRPCS1 ;DO RECALIBRATE COMMAND
74 016212 105777 164234          TSTB    SRPDS         ;DID DRIVE READY RESET DURING RECAL ?
75 016216 100011          BPL      10#          ;BR IF YES
76 016220 012737 000013 002272  MOV      #13,ERRMSK   ;LOG 'DRIVE READY DIDN'T RESET ERROR' STATUS
77 016226 004737 013304   JSR      PC,RETRY     ;SEE IF WE HAD ANY ERRORS
78 016232 000421          BR       NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
79
80 016234 005737 002274          TST      RTYCNT       ;ELSE RETURN HERE
81 016240 001356          BNE      9#           ;DID WE HAVE AN ERROR ?
82
83 016242 012737 000007 002404 10# :  MOV      #RECAL,FUNCTN ;SET UP FOR THE DRIVER
84 016250 004737 010502          JSR      PC,DRIVER    ;AND DO THE RECAL AGAIN
85 016254 004737 013304          JSR      PC,RETRY     ;SEE IF WE HAD ANY ERRORS
86 016260 000406          BR       NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
87
88 016262 005737 002274          TST      RTYCNT       ;ELSE RETURN HERE
89 016266 001365          BNE      10#          ;DID WE HAVE AN ERROR ?
90 016270 000405          BR       BUFINI       ;BR IF YES
91
92 016272 005237 002270          NORUN:  INC      ERR TTL ;INCREMENT THE TOTAL ERROR COUNT
93 016276 012737 177777 002262  NORUN:  MOV      #-1,SOF SW ;LOAD THE FAILED STATUS
94 016304 012701 031460          BUFINI: MOV      #BEGBUF,R1 ;GET POINTER TO BEGINNING OF BUFFER SECTION
95 016310 005021          1# :   CLR      (R1)+      ;INITIALIZE ALL BUFFERS

```

M6

96 016312 020127 120072
97 016316 101774
98 016320 000207

CMP R1,#ENDBUF
BLOS 1\$
RTS PC

;DONE YET ?
;BR IF NO

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 016322
18 016322 010246
19 016324 010346
20 016326 010546
21 016330 012702 000024
22 016334 012703 002440
23 016340
24 016340 013700 002420
25 016344 104442
26 016346 010005
27 016350 103034
28 016352 011346
29 016354 011546
30 016356 166616 000002
31 016362 061623
32 016364 005302
33 016366 001375
34 016370 004737 010330
35 016374 005737 002432
36 016400 001406
37 016402 013702 002430
38 016406 061502
39 016410 010223
40 016412 005722
41 016414 010213
42
43
44
45 016416 022626
46 016420 012537 002422
47 016424 012537 002424
48 016430 012537 002426
49 016434 011537 002434
50 016440 000406
51
52
53
54 016442 005237 002420
55 016446 023737 002420 002012
56 016454 002731
57 016456
58 016456 012605
59 016460 012603
60 016462 012602
61 016464 000207

```

```

*****
; THIS MODULE CREATES THE TABLES WHICH ARE USED TO SELECT DEVICE PARAMETERS
; USED IN THIS PROGRAM. INCLUDED IN THE PARAMETER LIST ARE THE FOLLOWING ITEMS:
; BASE REGISTER ADDRESSES FOR THE DEVICE, VECTOR ADDRESS OF THE DEVICE,
; DETERMINATION VIA OPERATOR FOR THE FOLLOWING: CONTROLLER TYPE. IN ADDITION,
; DIALOGUE DETERMINES THE PRIORITY OF THE CONTROLLER, AND THE SELECTION OF A
; DRIVE NUMBER FOR TEST.
;*
; INPUTS ARE: NON:
;
; OUTPUTS ARE: RPADD, RHTYPE, RPBAE, RPCS3, DRIVE, RPCS1
;*
; THIS MODULE IS CALLED BY: THE INITIALIZATION CODE
*****

```

```

TABELD::
      MOV     R2,-(SP)      ;;PUSH R2 ON STACK
      MOV     R3,-(SP)      ;;PUSH R3 ON STACK
      MOV     R5,-(SP)      ;;PUSH R5 ON STACK
      MOV     #20,R2        ;R2 = ITERATION COUNT
      MOV     #RPCS1,R3     ;R3 = DATA SINK
1$:   MOV     UNIT,R0        ;GET HARDWARE P-TABLE
      TRAP    C$GPHRD
      MOV     R0,R5
      BCC    4$
      MOV     (R3),-(SP)    ;SAVE R3
      MOV     (R5),-(SP)    ;AND THE BASE ADDRESS
      SUB     2(SP),(SP)    ;DERIVE NEW ADDRESS
2$:   ADD     (SP),(R3)+    ;LOG IT IN NEW TABLE
      DEC     R2            ;COUNT LOGGING
      BNE    2$            ;R2 NOT ZERO, CONTINUE LOGGING
      JSR    PC,SIZE70     ;SEE IF RH70 IS PRESENT
      TST    RHTYPE        ;IS IT AN RH70 ?
      BEQ    3$            ;BR IF NO
      MOV     RHEXT,R2     ;GET RPBAE OFFSET
      ADD     (R5),R2      ;ADD BASE ADDRESS TO OFFSET
      MOV     R2,(R3)+    ;SAVE NEW RPBAE
      TST    (R2)+        ;ADD 2
      MOV     R2,(R3)     ;SAVE NEW RPCS3
3$:   CMP     (SP)+,(SP)+  ;DONE, RESTORE THE STACK
      MOV     (R5)+,RPADR  ;SAVE RPCS1 BASE ADDRESS
      MOV     (R5)+,RPVEC  ;SAVE INTERUPT VECTOR ADDRESS
      MOV     (R5)+,RPVEC+2 ;SAVE INTERUPT PRIORITY
      MOV     (R5),DRVNO   ;SETUP DRIVE NUMBER FOR UNIT N
      BR     5$           ;SKIP NEXT
4$:   INC     UNIT         ;TRY THE NEXT UNIT
      CMP     UNIT,L$UNIT  ;IS THIS THE LAST UNIT TO TRY ?
      BLT    1$           ;BR IF NO
5$:   MOV     (SP)+,R5     ;;POP STACK INTO R5
      MOV     (SP)+,R3     ;;POP STACK INTO R3
      MOV     (SP)+,R2     ;;POP STACK INTO R2
      RTS    PC           ;TAKE THE RETURN

```

```

1      .SBTTL  MAP SECTOR POSITION
2
3      ;*****
4      ;THIS MODULE IS USED WHEN A USER INPUTS A DEFECT VIA THE SECTOR WORD COUNT
5      ;MECHANISM AND THE DRIVE IS IN INTERLEAVED MODE.  THE DATA IS CONVERTED FROM
6      ;INTERLEAVED POSITION TO LINEAR (NON-INTERLEAVED) POSITION.
7      ;*
8      ;INPUTS ARE:  SECADD
9      ;
10     ;OUTPUTS ARE:  SECADD
11     ;*
12     ;THIS MODULE IS CALLED BY:  RELTIVE
13     ;*****
14
15 016466 105737 002244  MAPSEC: TSTB  INTLEV  ;INTERLEAVED SECTORS ?
16 016472 001420          BEQ    2#    ;BR IF NO
17 016474 010146          MOV    R1,-(SP) ;;PUSH R1 ON STACK
   016476 010246          MOV    R2,-(SP) ;;PUSH R2 ON STACK
18 016500 113701 002323  MOVB  PLTRK,R1 ;GET THE FIRST INTERLEAVED SECTOR
19 016504 013702 002402  MOV    SECADD,R2 ;SAVE THE SECTOR ADDRESS
20 016510 006301          ASL    R1    ;MULTIPLY BY TWO
21 016512 006302          ASL    R2    ;FOR THE THRESHOLD AND THE INPUT
22 016514 020201          CMP    R2,R1 ;SCALE THE USER INPUT
23 016516 103402          BLO    1#    ;IT JUST NEEDS DOUBLING, IF LOWER!
24 016520 162702 000061  SUB    #49.,R2 ;THIS IS A SECTOR ADDRESS => 25.
25 016524 010237 002402  1#:  MOV    R2,SECADD ;SAVE THE CORRECTED SECTOR ADDRESS
26 016530 012602          MOV    (SP)+,R2 ;;POP STACK INTO R2
   016532 012601          MOV    (SP)+,R1 ;;POP STACK INTO R1
27 016534 000207          2#:  RTS   PC    ;RETURN NOW!
  
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27

```

.SBTTL FIND DEFECT RELATIVE TO INDEX
;*****
;THIS MODULE ACCEPTS SECTOR ADR AND WORD COUNT INPUT REPRESENTING A RELATIVE
;DEFECT, WHICH IS CONVERTED TO ABSOLUTE DATA, RELATIVE ONLY TO INDEX.
;*
;INPUTS ARE: INTLEV, SECADD, RELWRD
;
;OUTPUTS ARE: RELWRD
;*
;THIS MODULE IS CALLED BY: INPUTD, HDSCAN, DASCAN
;*****
RELATIVE:
MOV RO,-(SP) ;PUSH RO ON STACK
MOV R1,-(SP) ;PUSH R1 ON STACK
MOV SECADD,-(SP) ;PUSH SECADD ON STACK
JSR PC,MAPSEC ;CORRECT THE SECTOR MAP NOW!
MOV RELWRD,RO ;GET THE USER INPUT TO MAKE IT RELATIVE
ADD @<G1*TD*G2*HDR*G3>,RO ;GET INFO TO PUT THE DEFECT
;AT LEAST IN SECTOR 0
MOV @<HDR*G3*DATA*G4>,R1 ;GET SIZE OF SECTOR
1$: DEC SECADD ;KEEP TABS OF THE ITERATION COUNT
BMI 2$ ;WHEN MINUS, END!
ADD R1,RO ;ADD ONE MORE SECTOR TO VALUE
BR 1$ ;AND HOUSEKEEP
2$: MOV RO,RELWRD ;LOG THE ABSOLUTE VALUE RELATIVE TO INDEX
MOV (SP)+,SECADD ;POP STACK INTO SECADD
MOV (SP)+,R1 ;POP STACK INTO R1
MOV (SP)+,RO ;POP STACK INTO RO
RTS PC

```

```

016536
016536 010046
016540 010146
016542 013746 002402
016546 004737 016466
016552 013700 002400
016556 062700 000225
016562 012701 000517
016566 005337 002402
016572 100402
016574 060100
016576 000773
016600 010037 002400
016604 012637 002402
016610 012601
016612 012600
016614 000207

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48

```
.SBTTL  INSERT ABSOLUTE DEFECT INTO TDR
;*****
;THIS MODULE INSERTS THE ABSOLUTE DATA FROM RELATIVE TO INDEX INTO THE
;EXISTING TRACK DESCRIPTOR (IF POSSIBLE).  IF THE SCANNER IS BEING USED AND THE
;NUMBER OF DEFECTS EXCEEDS 4, THE MFG'S BAD SECTOR FILE WILL AUTOMATICALLY
;BE UPDATED, ELSE AN ERROR MESSAGE WILL BE GENERATED.
;
;INPUTS ARE:  RELWRD, TDWRD3 - TDWRD6
;
;OUTPUTS ARE: RELWRD, TDWRD3 - TDWRD6
;
;THIS MODULE IS CALLED BY: INPUTD, HDSCAN, DASCAN, UPDSCR
;*****
```

```
INSERT:
MOV  RO,-(SP)      ;PUSH RO ON STACK
MOV  R1,-(SP)      ;PUSH R1 ON STACK
MOV  #4,R0         ;GET THE # OF DEFECT WORDS TO CHECK IN TD
MOV  #TDWRD3,R1    ;FIRST ENTRY OF RECORD
1#:  CMP  (R1),#NULL ;IS THE TRACK DESCRIPTOR FULL ?
     BEQ  2#        ;BR IF NO
     DEC  R0         ;ONE LESS AVAILABLE ENTRY TO-GO
     BGT  1#        ;LOOK AGAIN
     BR   4#        ;NO ROOM AT THE INN - WE'VE GOT TROUBLE.
2#:  MOV  #-1,TMPBSF ;FLAG THAT DATA IS FOR TD, NOT BSF
     TST  TDWRD3    ;IS DATA VALID ?
     BEQ  13#       ;BR IF NO
     MOV  #4,R0     ;GET THE # OF DEFECT WORDS TO CHECK IN TD
     MOV  #TDWRD3,R1 ;GET POINTER TO BEGINNING OF DEFECTS
3#:  ADD  #<DS/2>,RELWRD ;ADD HALF A DEFECT TO ABSOLUTE VALUE
     CMP  RELWRD,(R1)+ ;DOES NEW DEFECT GO BEFORE THIS OLD DEFECT ?
     BLO  9#        ;BR IF YES
     SUB  -2(R1),RELWRD ;MAKE NEW DEFECT RELATIVE TO PREVIOUS DEFECT
     ADD  #<DS>,RELWRD ;ADD A DEFECT SKIP TO THE NEW DEFECT
     DEC  R0         ;DONE LOOKING YET ?
     BR   3#        ;BR IF NO
     ;SETUP TO PUT THIS NEW DEFECT INTO THE
     ;BAD SECTOR FILE BUFFER.
4#:  MOV  DESCYL,TMPBSF ;GET CYLINDER ADDRESS.
     MOV  SECADD,TMPBSF+2 ;SECTOR ADDRESS AND
     MOV  DESTRK+1,TMPBSF+3 ;TRACK ADDRESS.
     MOV  DESTRK+1,R0 ;GET TRACK ADDRESS
     PRINT 'WARNING ** CYL XXX.. TRK XX. HAS 4
     ; DEFECTS IN TRACK DESCRIPTOR'
MOV  RO,-(SP)
MOV  DESCYL,-(SP)
MOV  #WARN,-(SP)
MOV  #FRMT61,-(SP)
MOV  #4,-(SP)
MOV  SP,R0
TRAP C:PNTF
ADD  #12,SP
46:  CMP  TMPBSF,LSCYL ;SEE WHAT CYLINDER ADDRESS IS BEING FLAGGED.
47:  BHI  6#          ;BRANCH IF FE CYLINDER(S).
48:  BLO  5#          ;BRANCH IF ANY USER CYLINDER, ELSE
```

```
016616 010046
016616 010146
016620 010146
016622 012700 000004
016626 012701 031420
016632 022127 140000
016636 001403
016640 005300
016642 003373
016644 000430
016646 012737 177777 031460
016654 005737 031420
016660 001564
016662 012700 000004
016666 012701 031420
016672 062737 000066 002400
016700 023721 002400
016704 103513
016706 166137 177776 002400
016714 062737 000154 002400
016722 005300
016724 003365
016726 013737 002312 031460
016734 013737 002402 031462
016742 113737 002315 031463
016750 113700 002315
016754 010046
016756 013746 002312
016762 012746 006206
016766 012746 005055
016772 012746 000004
016776 010600
017000 104417
017002 062706 000012
017006 023737 031460 002222
017014 101013
017016 103404
```

```

49 017020 123737 031463 002224      CMPB   TMPBSF+3,LSTRK  ;SEE IF WE ARE FLAGGING THE BSF AREA,
50 017026 001406                      BEQ    6$              ;BRANCH IF SO, ELSE
51 017030 005737 002166      5$:   TST    SCANR      ;SEE IF SCAN MODE IS ENABLED,
52 017034 001014                      BNE    7$              ;BRANCH IF SO, ELSE
53 017036 005737 002202      TST    MODBY         ;MUST BE MODIFY MODE, SO CHECK TO SEE IF MODIFIING
54                                     ;BY WORD COUNT OR BY INDEX MODE,
55 017042 001423                      BEQ    8$              ;BRANCH IF WORD COUNT MODE, ELSE
56 017044                                6$:   ;PRINT '          INPUT DATA REJECTED'
57 017044 012746 005021      MOV    #FRMT60,-(SP)
   017050 012746 000001      MOV    #1,-(SP)
   017054 010600      MOV    SP,R0
   017056 104417      TRAP   C#PNTF
   017060 062706 000004      ADD    #4,SP
58 017064 000462      BR     13$
59 017066                                7$:   ;PRINT '          ANOTHER DEFECT WAS FOUND
60                                     ;          XXXXX. WORDS FROM INDEX'
61 017066 013746 002400      MOV    RELWRD,-(SP)
   017072 012746 005443      MOV    #FRMT65,-(SP)
   017076 012746 000002      MOV    #2,-(SP)
   017102 010600      MOV    SP,R0
   017104 104417      TRAP   C#PNTF
   017106 062706 000006      ADD    #6,SP
62 017112                                8$:   ;PRINT '          BAD SECTOR FILE WILL BE UPDATED'
63 017112 012746 005534      MOV    #FRMT66,-(SP)
   017116 012746 000001      MOV    #1,-(SP)
   017122 010600      MOV    SP,R0
   017124 104417      TRAP   C#PNTF
   017126 062706 000004      ADD    #4,SP
64 017132 000437      BR     13$
65
66 017134 162737 000066 002400  9$:   SUB    #<DS/2>,RELWRD ;SUBTRACT A HALF DEFECT FROM ABSOLUTE VALUE
67 017142 005741                      TST    -(R1)          ;BACK THE POINTER UP BY ONE WORD
68 017144 020027 000001      CMP    R0,#1         ;INSERT AFTER THREE DEFECTS?
69 017150 001416                      BEQ    12$           ;IF MATCH, YES
70 017152 020027 000002      CMP    R0,#2         ;INSERT AFTER TWO DEFECTS?
71 017156 001411                      BEQ    11$           ;IF MATCH, YES
72 017160 020027 000003      CMP    R0,#3         ;INSERT AFTER ONE DEFECT?
73 017164 001403                      BEQ    10$           ;IF MATCH, YES
74 017166 016161 000004 000006      MOV    4(R1),6(R1)   ;SHUFFLE THE DATA DOWN
75 017174 016161 000002 000004 10$:   MOV    2(R1),4(R1)   ;TO MAKE ROOM FOR THE NEW DEFECT
76 017202 011161 000002      11$:   MOV    (R1),2(R1)   ;KEEP SHUFFLING DOWN
77 017206 013711 002400      12$:   MOV    RELWRD,(R1)  ;LOG THE NEW DEFECT DATA NOW!
78 017212 005300                      DEC    R0            ;WAS THIS LAST ENTRY IN TD ?
79 017214 001406                      BEQ    13$           ;BR IF YES
80 017216 026127 000002 140000      CMP    2(R1),#NULL   ;NEXT WORD NULL ?
81 017224 001402                      BEQ    13$           ;BR IF YES
82 017226 161161 000002      SUB    (R1),2(R1)   ;ADJUST NEXT DEFECT TO MAKE IT RELATIVE TO
83                                     ;THE NEW DEFECT
84 017232                                13$:   ;POP STACK INTO R1
   017232 012601      MOV    (SP)+,R1
   017234 012600      MOV    (SP)+,R0
85 017236 000207      RTS    PC           ;NOW RETURN TO MAIN

```

```

1          .SBTTL TRACK DESCRIPTOR RECORD DUMP
2
3          ;*****
4          ;THIS MODULE PRODUCES THE TRACK DESCRIPTOR FILE ON THE LOCAL CONSOLE PRINTER.
5          ;IT IS INVOKED ANYTIME A USER REQUESTS A TRACK DESCRIPTOR DUMP.
6          ;*
7          ;INPUTS ARE: TEXT, TDWRD1 - TDWRD6
8          ;
9          ;OUTPUTS ARE: NONE
10         ;*
11         ;THIS MODULE IS CALLED BY: INPUTD, READTD
12         ;*****
13
14         TDDUMP:
15         017240 010146          MOV R1,-(SP)          ;;PUSH R1 ON STACK
16         017242 010246          MOV R2,-(SP)          ;;PUSH R2 ON STACK
17         017244 005737 002172    TST LIST             ;IS LIST OPTION ENABLED ?
18         017250 001404          BEQ 1$               ;BR IF NO
19         017252 023727 002254 000001  CMP DEFCNT,#1        ;WAS FIRST DEFECT LISTED ALL READY ?
20         017260 002022          BGE 2$               ;BR IF YES
21         017262          1$: MOV TEXT,-(SP)          ;PRINT THE APPROPRIATE MESSAGE
22         017266 013746 002360    MOV #FRMTT,-(SP)
23         017272 012746 002661    MOV #2,-(SP)
24         017276 012746 000002    MOV SP,R0
25         017276 010600          TRAP C#PNTF
26         017300 104417          ADD C#PNTF
27         017302 062706 000006    ADD #6,SP
28
29         ;PRINT 'WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5
30
31         017306 012746 004406    MOV #FRMT35,-(SP)
32         017312 012746 000001    MOV #1,-(SP)
33         017316 010600          MOV SP,R0
34         017320 104417          TRAP C#PNTF
35         017322 062706 000004    ADD #4,SP
36         017326 013746 031416    2$: MOV TDWRD2,-(SP) ;GET THE 2ND WORD OF THE TD (TRK/SEC ADRS)
37         017332 011646          MOV (SP),-(SP)      ;SAVE 2ND TD WORD
38         017334 105066 000003    CLRB 3(SP)          ;GET RID OF HI BYTE (EXTRACT SECTOR)
39         017340 011646          MOV (SP),-(SP)      ;SAVE 2ND TD WORD
40         017342 105066 000002    CLRB 2(SP)          ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
41         017346 000366 000002    SWAB 2(SP)          ;MAKE IT THE LOW BYTE DATA.
42         017352 013746 031414    MOV TDWRD1,-(SP)   ;GET THE FIRST WORD OF TD (CYLINDER ADDRESS)
43         017356 011646          MOV (SP),-(SP)      ;SAVE 1ST TD WORD
44         017360 042766 170000 000002  BIC #170000,2(SP)   ;STRIP THE STATUS BITS
45
46         ;PRINT TD WORD#1, CYL, TD WORD#2 & TRK/SEC DATA
47
48         017366 012646          MOV (SP)+,-(SP)
49         017370 012646          MOV (SP)+,-(SP)
50         017372 012646          MOV (SP)+,-(SP)
51         017374 012646          MOV (SP)+,-(SP)
52         017376 012646          MOV (SP)+,-(SP)
53         017400 012746 004515    MOV #FRMT36,-(SP)
54         017404 012746 000006    MOV #6,-(SP)
55         017410 010600          MOV SP,R0
56         017412 104417          TRAP C#PNTF
57         017414 062706 000016    ADD #16,SP
58         017420 012701 031420    MOV #TDWRD3,R1      ;GET THE DEFECT FILE
59         017424 012702 000004    MOV #4,R2           ;AND THE MAXIMUM NUMBER OF DEFECTS
60         017430 022127 140000    3$: CMP (R1)+,#NULL  ;IS IT NULL DATA ?
61         017434 001413          BEQ 4$              ;BR IF YES
    
```

```

38
39 017436 016146 177776          MOV    -2(R1),-(SP)      ;PRINT THE DEFECT DATA
   017442 012746 003640          MOV    #FRMT14, -(SP)
   017446 012746 000002          MOV    #2, -(SP)
   017452 010600                  MOV    SP,R0
   017454 104417                  TRAP   C#PNTF
   017456 062706 000006          ADD    #6,SP
40 017462 000412                  BR     5#
41 017464                          4#:
42 017464 016146 177776          MOV    -2(R1),-(SP)      ;PRINT ' 140000 '
   017470 012746 003651          MOV    #FRMT15, -(SP)
   017474 012746 000002          MOV    #2, -(SP)
   017500 010600                  MOV    SP,R0
   017502 104417                  TRAP   C#PNTF
   017504 062706 000006          ADD    #6,SP
43 017510 005302                  5#:
44 017512 003346                  DEC    R2                ;ONE LESS ITERATION TO-GO
45                                BGT   3#                ;DO UNTIL = 0!
46 017514 012746 002656          MOV    #CRLF, -(SP)      ;CR-LF
   017520 012746 000001          MOV    #1, -(SP)
   017524 010600                  MOV    SP,R0
   017526 104417                  TRAP   C#PNTF
   017530 062706 000004          ADD    #4,SP
47 017534 012602                  MOV    (SP)+,R2          ;;POP STACK INTO R2
   017536 012601                  MOV    (SP)+,R1          ;;POP STACK INTO R1
48 017540 000207                  RTS    PC                ;TAKE THE RETURN

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26

```
.SBTTL REGISTER BUFFER MODULE
;*****
;THIS MODULE SAVE AN IMAGE OF THE RP REGISTERS. IT IS USED FOR DIAGNOSTIC
;PURPOSES SUBSEQUENT TO A DISK COMMAND.
;*
;INPUTS ARE:  RPCS1 - RPCS3.
;
;OUTPUTS ARE:  REG+0 - REG+52
;*
;THIS MODULE IS CALLED BY: ERRO
;*****
```

```
SAVRPR:
      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
      MOV      @RPCS1,R2     ;AND THE TRANSFER ADDRESSES
      MOV      @REG,R1       ;OF THE SOURCE AND SINK BUFFERS
      MOV      @20.,R3       ;GET THE ITERATION COUNT
1$:   MOV      @R2)+,(R1)+   ;NOW LOG THE DATA
      DEC      R3            ;ONE LESS REGISTER TO GO
      BGT      1$           ;IF >0, KEEP GOING
      TST      RHTYPE       ;WHICH CONTROLLER?
      BEQ      2$           ;IF ZERO, IT'S AN RH11
      MOV      @R2)+,(R1)+   ;LOG RPBAE
      MOV      @R2)+,(R1)+   ;LOG RPCS3
2$:   MOV      (SP)+,R3      ;;POP STACK INTO R3
      MOV      (SP)+,R2      ;;POP STACK INTO R2
      MOV      (SP)+,R1      ;;POP STACK INTO R1
      RTS      PC           ;;RETURN TO MAIN FOR ERROR REPORT
```

```
017542 010146
017544 010246
017546 010346
017550 012702 002440
017554 012701 002514
017560 012703 000024
017564 013221
017566 005303
017570 003375
017572 005737 002432
017576 001402
017600 013221
017602 013221
017604 012603
017606 012602
017610 012601
017612 000207
```

```

1      .SBTTL  HEADER VERIFY MODULE
2
3      ;*****
4      ;THIS MODULE DOES THE HEADER AND S.D. VERIFICATION FOR A VERIFY MODE OF
5      ;OPERATION.  THE TRACK DESCRIPTOR IS READ, AND CALCULATIONS ARE PERFORMED
6      ;ON THE TD WORDS FOUND.  THESE CALCULATIONS ARE THEN TABLED AND COMPARED
7      ;WITH THE ACTUAL S.D. WORDS FOUND ON A SPECIFIED TRACK.  ANY MISMATCH WILL
8      ;RESULT IN SEVERAL RETRY ATTEMPTS, THEN FINALLY AN ERROR MESSAGE.
9      ;*
10     ;INPUTS ARE:  HDRBLK+0 - HDRBLK+300., SEC50, INTLEV, RTYCNT, TDWRD3 - TDWRD6
11     ;
12     ;OUTPUTS ARE:  DEFCNT, ERRMSK, DEF1 - DEF4, LINBUF+0 - LINBUF+300.
13     ;*
14     ;THIS MODULE IS CALLED BY:  MAIN
15     ;*****
16
17 017614 012700 032464  VERIFY:  MOV      #HDRBLK,R0      ;GET POINTER TO BEGINNING OF EXPECTED DATA
18 017620 105737 002244      TSTB     INTLEV      ;INTERLEAVED DRIVE ?
19 017624 001404      BEQ      1$          ;BR IF NO
20 017626 004737 011630      JSR      PC,CHABUF   ;RE-SHUFFLE THE DATA FOR A LINEAR COMPARE
21 017632 012700 034744      MOV      #LINBUF,R0  ;GET POINTER TO BEGINNING OF EXPECTED DATA
22 017636 113702 002322  1$:  MOVB     SEC50,R2   ;# OF SECTORS ON TRACK
23 017642 012701 033614      MOV      #TDSBLK,R1 ;GET POINTER TO BEGINNING OF RECEIVED DATA
24 017646 012704 000006  2$:  MOV      #<HDR-2>,R4 ;# OF WORDS IN HEADER TO VERIFY
25 017652 022021  3$:  CMP      (R0)+,(R1)+ ;DOES EXPECTED DATA MATCH RECEIVED DATA ?
26 017654 001007      BNE      4$          ;BR IF NO
27 017656 005304      DEC      R4          ;DONE COMPARING THIS SECTOR YET ?
28 017660 001374      BNE      3$          ;BR IF NO
29 017662 005302      DEC      R2          ;DONE COMPARING THIS TRACK YET ?
30 017664 002370      BGE      2$          ;BR IF NO
31 017666 005037 002272  CLR      ERRMSK     ;RESET NO ERROR STATUS AND SETUP CONDITION CODE
32                                     ;IN PSW FOR RETURN
33 017672 000207      RTS      PC
34
35 017674 014037 002276  4$:  MOV      -(R0),EXPCTD ;SAVE THE EXPECTED DATA
36 017700 014137 002300      MOV      -(R1),RECVD  ;AND THE RECEIVED DATA
37 017704 012737 000010 002272  MOV      #10,ERRMSK   ;LOG 'FORMAT VERIFY ERROR' STATUS AND
38                                     ;SETUP CONDITION CODE IN PSW FOR RETURN
39 017712 000207      RTS      PC

```

```

1      .SBTTL  READ MFG BAD SECTOR FILE MODULE
2
3
4      ;*****
5      ;THIS MODULE DOES A READ OF THE MANUFACTURES SECTION (SECTORS 0,2,4,6,8) OF
6      ;THE BAD SECTOR FILE (DEC144). IF THE READ IS UNSUCCESSFUL, THE PROGRAM
7      ;ASSUMES THAT THE BSF WAS NEVER CREATED AND MARKS STATUS AS SUCH. IF DATA IS
8      ;FOUND FROM ONE OF THE FIVE COPIES OF THE MFG'S BAD SECTOR FILE, THE
9      ;INFORMATION IS STORED FOR A LATER REWRITE.
10     ;*
11     ;INPUTS ARE:  MFGBUF, RDDAT, RTYCNT
12     ;
13     ;OUTPUTS ARE:  SUPRSS, DESCYL, DESTRK, BUSADR, FUNCTN, NEGWRD, ERRMAX, CREATE
14     ;*
15     ;THIS MODULE IS CALLED BY: MAIN
16     ;*****
17 017714 012737 177777 002252  RDBSF:  MOV      #-1,SUPRSS      ;SUPPRESS THE ERROR MESSAGE OUTPUT
18 017722 010046          002312      MOV      RO,-(SP)        ;;PUSH RO ON STACK
19 017724 013746 002312          MOV      DESCYL,-(SP)    ;;PUSH DESCYL ON STACK
20 017730 013746 002314          MOV      DESTRK,-(SP)   ;;PUSH DESTRK ON STACK
21 017734 013746 002404          MOV      FUNCTN,-(SP)  ;;PUSH FUNCTN ON STACK
22 017740 013737 002222 002312      MOV      LSCYL,DESCYL  ;GET THE BAD SECTOR FILE (LAST) CYLINDER,
23 017746 112737 000000 002314      MOVVB   #0,DESTRK      ;SECTOR AND
24 017754 113737 002224 002315      MOVVB   LSTRK,DESTRK+1 ;TRACK ADDRESSES.
25 017762 012737 031464 002234  1#:  MOV      #MFGBUF,BUSADR ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
26 017770 012737 000071 002404      MOV      #RDDAT,FUNCTN ;SET UP FOR A READ DATA COMMAND
27 017776 012737 000400 002376      MOV      #256.,NEGWRD  ;WORD COUNT FOR THE ENTIRE SECTOR
28 020004 004737 010502          JSR      PC,DRIVER     ;EXECUTE THE OPERATION
29 020010 004737 013304          JSR      PC,RETRY     ;SEE IF WE HAD ANY ERRORS
30 020014 000404          BR       2#          ;RETURN HERE IF EXCEEDED RETRY LIMIT
31 020016 005737 002274          TST     RTYCNT        ;ELSE RETURN HERE
32 020022 001357          BNE     1#          ;DID WE HAVE AN ERROR ?
33 020024 000425          BR       3#          ;BR IF YES
34 020026 062737 000002 002314  2#:  ADD     #2,DESTRK     ;INCREMENT TO NEXT COPY OF MFG BSF
35 020034 123727 002314 000012      CMPB    DESTRK,#10.   ;DID WE TRY READING ALL COPIES OF MFG BSF ?
36 020042 002747          BLT     1#          ;BR IF NO
37 020044 012746 006206          MOV     #WARN,-(SP)  ;PRINT 'WARNING ** BAD SECTOR FILE DATA IS
38 020050 012746 005342          MOV     #FRMT64,-(SP); CORRUPT, IT WILL BE INITIALIZED'
39 020054 012746 000002          MOV     #2,-(SP)
40 020060 010600          MOV     SP,RO
41 020062 104417          TRAP   C#PNTF
42 020064 062706 000006          ADD     #6,SP
43 020070 012700 031464          MOV     #MFGBUF,RO   ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
44 020074 004737 020550          JSR     PC,CRE144    ;CREATE A NULLED MFG BSF BUFFER
45 020100 012700 031464          ;NOW THAT WE HAVE A GOOD READ OR AN INITIALIZED
46 020104 022027 000105          ;BUFFER, LET'S SEE IF THE DATA IS VALID.
47 020110 001346          ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
48 020112 022027 000105          MOV     (RO)+,#105   ;LOOK FOR THE PACK S.N.
49 020116 001343          BNE     2#          ;NO MATCH, REWRITE THE MFG BSF
50 020120 005720          CMP     (RO)+,#105   ;LOOK FOR THE PACK S.N.
51 020122 001341          BNE     2#          ;NO MATCH, REWRITE THE MFG BSF
52          TST     (RO)+   ;THIS WORD ALWAYS 0!
53          BNE     2#   ;IF NOT, FILE IS BAD

```


50	020124	005720	TST	(R0)+	;IS THIS AN ALIGNMENT PACK ?
51	020126	001337	BNE	2*	;BR IF YES
52	020130	004737	JSR	PC,RESET	;RESET ANY ERRORS
53	020134	012637	MOV	(SP)+,FUNCTN	::POP STACK INTO FUNCTN
	020140	012637	MOV	(SP)+,DESTRK	::POP STACK INTO DESTRK
	020144	012637	MOV	(SP)+,DESCYL	::POP STACK INTO DESCYL
	020150	012600	MOV	(SP)+,R0	::POP STACK INTO R0
54	020152	005037	CLR	SUPRSS	;ALLOW ERROR MESSAGE TO BE OUTPUT
55	020156	000207	RTS	PC	;RETURN TO MAIN

```

1      .SBTTL WRITE MFG/USR BAD SECTOR FILE MODULE
2
3      ;*****
4      ;THIS MODULE CREATES AND WRITES THE BAD SECTOR FILE ON THE LAST USER
5      ;CYLINDER/TRACK ON THE DISK.
6      ;*
7      ;INPUTS ARE:   WRTDAT, WRDAT, MFGBUF, DBUFF, CREATE
8      ;
9      ;OUTPUTS ARE:  DESCYL, DESTRK, FUNCTN, ERRFNC, MFG144, BUSADR, NEGWRD, ERRMSK,
10     ;              SEC50
11     ;*
12     ;THIS MODULE IS CALLED BY: MAIN
13     ;*****
14
15     WRTBSF:
16     020160 013746 002312      MOV     DESCYL,-(SP)      ;PUSH DESCYL ON STACK
17     020164 013746 002314      MOV     DESTRK,-(SP)     ;PUSH DESTRK ON STACK
18     020170 004737 020414      JSR     PC, SORT        ;SORT THE MFG BSF ENTRIES
19     020174 012700 036074      MOV     #DBUFF,RO       ;GET POINTER TO BEGINNING OF BUFFER
20     020200 004737 020550      JSR     PC, CRE144      ;CREATE THE USR BSF BUFFER
21     020204 013737 002222 002312  MOV     LSCYL,DESCYL    ;GET THE BAD SECTOR FILE (LAST) CYLINDER,
22     020212 112737 000000 002314  MOVVB   #0,DESTRK      ;SECTOR AND
23     020220 113737 002224 002315  MOVVB   LSTRK,DESTRK+1 ;TRACK ADDRESSES.
24     020226 012737 031464 002234 1#:  MOV     #MFGBUF,BUSADR ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
25     020234 012737 000061 002404 2#:  MOV     #WRTDAT,FUNCTN ;SET UP FOR A WRITE DATA FUNCTION
26     020242 012737 005625 002260  MOV     #WRDAT,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
27     020250 012737 000400 002376  MOV     #256.,NEGWRD   ;AND THE SECTOR WORD COUNT
28     020256 004737 010502      JSR     PC,DRIVER      ;EXECUTE THE OPERATION
29     020262 004737 013304      JSR     PC,RETRY      ;SEE IF WE HAD ANY ERRORS
30     020266 000403      BR      3#            ;RETURN HERE IF EXCEEDED RETRY LIMIT
31     ;ELSE RETURN HERE
32     020270 005737 002274      TST     RTYCNT         ;DID WE HAVE AN ERROR ?
33     020274 001357      BNE     2#            ;BR IF YES
34     020276 105237 002314 3#:  INCB   DESTRK         ;NEXT SECTOR
35     020302 123727 002314 000012  CMPB   DESTRK,#10.    ;DONE WITH 16 & 18 BIT MFG BSF YET ?
36     020310 002010      BGE     4#            ;BR IF YES
37     020312 022737 036074 002234  CMP     #DBUFF,BUSADR ;DID WE JUST WRITE A 18 BIT MFG BSF ?
38     020320 001742      BEQ     1#            ;BR IF YES
39     020322 012737 036074 002234  MOV     #DBUFF,BUSADR ;GET POINTER TO BEGINNING OF 18 BIT BSF BUFFER
40     020330 000741      BR      2#
41     ;START WRITTING USR 16 & 18 BIT BSF
42     020332 012737 036074 002234 4#:  MOV     #DBUFF,BUSADR ;GET POINTER TO BEGINNING OF USR BSF BUFFER
43     020340 012737 000400 002376  MOV     #256.,NEGWRD  ;AND THE SECTOR WORD COUNT
44     020346 004737 010502      JSR     PC,DRIVER      ;EXECUTE THE OPERATION
45     020352 004737 013304      JSR     PC,RETRY      ;SEE IF WE HAD ANY ERRORS
46     020356 000403      BR      5#            ;RETURN HERE IF EXCEEDED RETRY LIMIT
47     ;ELSE RETURN HERE
48     020360 005737 002274      TST     RTYCNT         ;DID WE HAVE AN ERROR ?
49     020364 001362      BNE     4#            ;BR IF YES
50     020366 105237 002314 5#:  INCB   DESTRK         ;NEXT SECTOR
51     020372 123737 002314 002322  CMPB   DESTRK,SEC50  ;DONE WITH USR 16 & 18 BIT BSF YET ?
52     020400 003754      BLE     4#            ;BR IF NO
53     020402 012637 002314      MOV     (SP)+,DESTRK  ;POP STACK INTO DESTRK
54     020406 012637 002312      MOV     (SP)+,DESCYL ;POP STACK INTO DESCYL
55     020412 000207      RTS     PC
    
```

```

1      .SBTTL  SORT BAD SECTOR FILE
2
3      ;*****
4      ;THIS ROUTINE WILL SORT OUT DUPLICATE ENTRIES AND PUT THE BAD SECTOR FILE
5      ;IN ASCENDING ORDER.
6      ;CALL
7      ;      JSR      PC,SORT      ;CALL THE ROUTINE
8      ;*****
9
10     SORT:  NOP
11     020414 000240      MOV      #MFGBUF+8.,R1      ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
12     020416 012701 031474      ;TO GET OVER 2 S/N WRDS AND 2 ALL 0'S WRDS.
13     020422 012703 032462      MOV      #ENDBSF,R3      ;GET POINTER TO END OF MFG BSF BUFFER
14     020426 005711      TST      (R1)      ;DONE CHECKING LIST YET ?
15     020430 100446      BMI      5$      ;BR IF YES
16     020432 010102      1$:  MOV      R1,R2      ;RESTORE BASE POINTER TO R2
17     020434 000422      BR      3$
18     020436 021211      2$:  CMP      (R2),(R1)      ;COMPARE THE CYLINDER NUMBER
19     020440 103427      BLO     4$      ;SWAP ENTRIES
20     020442 101017      BHI     3$      ;INCREMENT POINTERS
21     020444 126261 000003 000003      CMPB    3(R2),3(R1)      ;COMPARE THE TRACK NUMBER
22     020452 103422      BLO     4$      ;SWAP ENTRIES
23     020454 101012      BHI     3$      ;INCREMENT POINTERS
24     020456 126261 000002 000002      CMPB    2(R2),2(R1)      ;COMPARE THE SECTOR NUMBER
25     020464 103415      BLO     4$      ;SWAP ENTRIES
26     020466 101005      BHI     3$      ;INCREMENT POINTERS
27     020470 012712 177777      MOV      #-1,(R2)      ;GET RID OF DUPLICATE CYLINDER AND
28     020474 012762 177777 000002      MOV      #-1,2(R2)      ;DUPLICATE TRACK/SECTOR ENTRIES.
29     020502 022222      3$:  CMP      (R2)+,(R2)+      ;UPDATE SEARCH POINTER TO NEXT CYLINDER
30     020504 020203      CMP      R2,R3      ;SEE IF THIS IS THE END OF BSF BUFFER.
31     020506 103753      BLO     2$      ;BRANCH IF NOT, ELSE
32     020510 022121      CMP      (R1)+,(R1)+      ;UPDATE BASE POINTER TO NEXT CYLINDER
33     020512 020103      CMP      R1,R3      ;SEE IF THIS IS THE END OF BSF BUFFER.
34     020514 103746      BLO     1$      ;BRANCH IF NOT, ELSE
35     020516 000413      BR      5$      ;EXIT
36
37     020520 011146      4$:  MOV      (R1),-(SP)      ;SAVE THE CURRENT LO CYLINDER ENTRY
38     020522 016146 000002      MOV      2(R1),-(SP)      ;SAVE THE CURRENT LO TRACK/SECTOR ENTRY
39     020526 011211      MOV      (R2),(R1)      ;SWITCH THE LO CYLINDER ENTRY
40     020530 016261 000002 000002      MOV      2(R2),2(R1)      ;SWITCH THE LO TRACK/SECTOR ENTRY
41     020536 012662 000002      MOV      (SP)+,2(R2)      ;LOAD THE HI CYLINDER ENTRY
42     020542 012612      MOV      (SP)+,(R2)      ;LOAD THE HI TRACK/SECTOR ENTRY
43     020544 000756      BR      3$
44
45     020546 000207      5$:  RTS      PC      ;EXIT

```

```

1
2
3
4
5
6
7
8
9
10
11
12 020550 010001
13 020552 062701 001000
14 020556 012720 000105
15 020562 012720 000105
16 020566 005020
17 020570 005020
18 020572 012720 177777
19 020576 020001
20 020600 103774
21 020602 000207
22

;*****
;THIS MODULE CREATES THE MANUFACTURES OR USER SECTION OF THE BAD SECTOR FILE
;(DEC 144 FILE), WHICH IS WRITTEN TO THE LAST USER CYLINDER/TRACK ON THE DISK.
;*
;INPUTS ARE: RO = POINTER TO BEGINNING OUTPUT BUFFER AREA
;
;OUTPUTS ARE: BUFFER POINTED TO BY RO
;*
;THIS MODULE IS CALLED BY: RDBSF, WRTBSF
;*****
CRE144: MOV RO,R1 ;GET THE START OF THE BUFFER
        ADD #<256.*2>,R1 ;SAVE END OF BUFFER IN R1
        MOV #105,(RO)+ ;CREATE A "HDA SERIAL NUMBER"
        MOV #105,(RO)+ ;TO SATISFY ALL OPERATING SYSTEMS
        CLR (RO)+ ;THIRD WORD ALWAYS = 0!
        CLR (RO)+ ;FOURTH WORD = 0 (<> ALIGNMENT PACK)
1$: MOV #-1,(RO)+ ;NULL THE BAD SECTOR INFORMATION
     CMP RO,R1 ;FILE DONE?
     BLO 1$ ;NOT IF RO < SAID NUMBER
     RTS PC ;OK, TAKE RETURN
    
```

```

1      .SBTTL ADDRESSING ROUTINES
2
3      ;;*****
4      ;THIS MODULE IS USED IN THE SCANNING PROCESS TO PROVIDE ADDRESSES TO THE EXEC
5      ;WHEN A SECOND OR CONSECUTIVE ITERATION IS NEEDED DURING A SCAN OPERATION.
6      ;THE ADDRESSES ARE LOCATED IN THE BUFFER "DBUFF" AND ARE PLACED THERE AS A
7      ;RESULT OF A PREVIOUS SCAN OPERATION.
8      ;*
9      ;INPUTS ARE:   TBLPTR
10     ;
11     ;OUTPUTS ARE:  DESCYL, DESTRK
12     ;*
13     ;THIS MODULE IS CALLED BY: MAIN
14     ;;*****
15
16 020604 013701 002256   GETNEX: MOV     TBLPTR,R1      ;SETUP DEFECT BUFFER POINTER
17 020610 010146                MOV     R1,-(SP)      ;;PUSH R1 ON STACK
18 020612 012137 002312   MOV     (R1),DESCYL  ;GET THE CYLINDER ADDRESS
19 020616 042737 170000 002312   BIC     @170000,DESCYL ;REMOVE ANY STATUS BITS
20 020624 012137 002314   MOV     (R1),DESTRK  ;AND LOAD IT NOW
21 020630 105037 002314   CLR     DESTRK      ;EXTRACT SECTOR FROM DISK ADDRESS
22 020634 062737 000016 002256   ADD     @<7*2>,TBLPTR ;GET NEXT ENTRY IN BUFFER
23 020642 012601                MOV     (SP),R1      ;;POP STACK INTO R1
24 020644 000207                ;:     RTS     PC
25
26     ;;*****
27     ;THIS MODULE CONTROLS SPIRAL READS OR WRITES. IT INCREMENTS THE DESIRED TRACK
28     ;ADDRESS UNTIL IT IS READY TO OVERFLOW INTO AN ILLEGAL ADDRESS. WHEN THIS IS
29     ;READY TO HAPPEN, THE DESIRED TRACK IS RESET TO 'MINTRK' AND THE DESIRED
30     ;CYLINDER IS INCREMENTED. THE CALLING MODULE MUST DETECT WHEN THE DESIRED
31     ;CYLINDER HAS OVERFLOWED.
32     ;*
33     ;VARIABLES:   DESCYL AND DESTRK ARE AFFECTED BY THIS MODULE.
34     ;
35     ;INPUTS ARE:  MINTRK, MAXTRK
36     ;
37     ;OUTPUTS ARE: DESCYL, DESTRK
38     ;*
39     ;THIS MODULE IS CALLED BY: MAIN
40     ;;*****
41
42 020646 105237 002315   SPIRAL: INCB   DESTRK+1  ;INCREMENT TO NEXT TRACK ADDRESS
43 020652 123737 002315 002220   CMPB   DESTRK+1,MAXTRK ;TRACK COUNT AT MAX?
44 020660 101405                BLOS   ;:      ;BR IF NO
45 020662 113737 002216 002315   MOVB   MINTRK,DESTRK+1 ;RESTORE MIN TRACK ADDRESS
46 020670 005237 002312   INC    DESTRK      ;INCREMENT TO NEXT CYLINDER ADDRESS
47 020674 000207                ;:     RTS     PC

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41

020676
020676 010146
020700 010246
020702 010346
020704 023727 031420 140000
020712 001452
020714 005737 031420
020720 001447
020722 012702 000004
020726 012701 031420
020732 012703 002302
020736 022127 140000
020742 001435
020744 005237 002254
020750 005723
020752 005302
020754 003370
020756 005737 002172
020762 001025
020764 005737 002170
020770 001022
020772 013701 031414
020776 042701 170000
021002 113702 031417

021006 010246
021010 010146
021012 012746 006206
021016 012746 005055
021022 012746 000004
021026 010600
021030 104417
021032 062706 000012
021036 005243
021040
021040 012603
021042 012602
021044 012601
021046 000207

```
*****  
; THIS MODULE IS USED TO CONTROL THE COUNTING OF DEFECTS ENCOUNTERED ON THE  
; RP07 HDA.  
; *  
; INPUTS ARE: FROM TDWRD3 - TDWRD6  
; *  
; OUTPUTS ARE: DEF1 - DEF4, TRACK AND DEFCNT (DEPENDENT UPON THE # OF DEFECTS  
; FOUND)  
; *  
; CALL  
; JSR PC,CNTDEF  
*****  
  
CNTDEF:  
MOV R1,-(SP) ; PUSH R1 ON STACK  
MOV R2,-(SP) ; PUSH R2 ON STACK  
MOV R3,-(SP) ; PUSH R3 ON STACK  
CMP TDWRD3,#NULL ; DEFECT INFORMATION ?  
BEQ 3# ; BR IF NO  
TST TDWRD3 ; IS IT VALID DATA ?  
BEQ 3# ; BR IF NO  
MOV #4,R2 ; FOUR RECORDS  
MOV #TDWRD3,R1 ; INPUT FROM HERE  
MOV #DEF1,R3 ; OUTPUT COUNT GOES HERE  
1# : CMP (R1),#NULL ; DATA OR UNUSED ?  
BEQ 2# ; IF MATCH, UNUSED  
INC DEFCNT ; INCLUDE THIS IN THE OVERALL COUNT  
TST (R3) ; BUMP THE POINTER  
DEC R2 ; ONE LESS ENTRY TO GO  
BGT 1# ; IF NOT ZERO, WE'RE NOT DONE  
TST LIST ; IS LIST OPTION ENABLED ?  
BNE 2# ; IF NOT ZERO, YES  
TST MODTD ; IS MODIFY TD MODE ENABLED ?  
BNE 2# ; BR IF YES  
MOV TDWRD1,R1 ; GET THE 1ST WORD OF TD (CYLINDER ADDRESS)  
BIC #170000,R1 ; STRIP THE STATUS BITS  
MOVB TDWRD2+1,R2 ; GET HI BYTE OF 2ND WORD OF TD (TRACK ADDRESS)  
; PRINT 'WARNING ** CYL XXX., TRK XX. HAS 4  
; DEFECTS IN TRACK DESCRIPTOR'  
  
MOV R2,-(SP)  
MOV R1,-(SP)  
MOV #WARN,-(SP)  
MOV #FRMT61,-(SP)  
MOV #4,-(SP)  
MOV SP,R0  
TRAP C#PNTF  
2# : ADD #12,SP  
3# : INC -(R3) ; INCLUDE THIS IN # DEFECTS/TRACK  
  
MOV (SP),R3 ; POP STACK INTO R3  
MOV (SP),R2 ; POP STACK INTO R2  
MOV (SP),R1 ; POP STACK INTO R1  
RTS PC ; RETURN TO THE USER
```

```

1
2
3
4
5
6
7
8 021050 012737 001167 002312 FETWO: MOV #631.,DESCYL ;GET THE SECOND FE CYLINDER ADDRESS
9 021056 005037 002314          CLR   DESTRK      ;RESET THE TRACK/SECTOR ADDRESS
10 021062 012701 031414          1$:  MOV   #TDWRD1,R1 ;SET UP A BUFFER POINTER
11 021066 013711 002312          MOV   DESCYL,(R1)  ;FIRST WORD OF TD
12 021072 052721 030000          BIS   #BIT13:BIT12,(R1) ;IT'S A TD - IN 16 BIT MODE
13 021076 013711 002314          MOV   DESTRK,(R1)  ;TRACK / SECTOR ADDRESS
14 021102 052721 000377          BIS   #377,(R1)    ;SECTOR ADDRESS = - 1
15 021106 012702 000004          MOV   #4,R2        ;A RECORD ITERATION COUNTER
16 021112 012721 140000          2$:  MOV   #NULL,(R1) ;NO DATA FOR THIS TD (LEAVE IT NULL)
17 021116 005302          DEC   R2           ;ONE LESS RECORD TO GO
18 021120 003374          BGT   2$          ;IF NOT ZERO, WE'RE NOT DONE
19 021122 004737 021152          JSR   PC,WRITTD   ;ATTEMPT TO WRITE THIS RECORD
20 021126 000207          RTS   PC           ;RETURN HERE IF EXCEEDED RETRY LIMIT
21
22 021130 105237 002315          INCB  DESTRK+1    ;NEXT TRACK PLEASE.....
23 021134 123727 002315 000037  CMPB  DESTRK+1,#31. ;LAST TRACK YET?
24 021142 101747          BLOS  1$          ;TAKE BRANCH IF NOT
25 021144 105337 002315          DECB  DESTRK+1    ;GET RID OF THE OVERFLOW ADDRESS
26 021150 000207          RTS   PC           ;TAKE THE USER RETURN

```

```

1      .SBTTL WRITE TRACK DESCRIPTOR MODULE
2
3      ;*****
4      ;THIS MODULE SIZES THE TD DATA AND DETERMINES IF THE TD ITSELF MUST BE
5      ;RELOCATED DUE TO THE PRESENCE OF A DEFECT WHERE THE TD WOULD NORMALLY BE
6      ;FOUND. IT ALSO DETERMINES IF HEADER #0 MUST BE RELOCATED DUE TO A DEFECT
7      ;BETWEEN THE TD FILE AND THE FIRST HEADER. MASK BITS ARE SET TO INFORM THE
8      ;DEVICE EITHER OF THESE CONDITIONS OCCUR.
9      ;*
10     ;INPUTS ARE: TDWRD1, TDWRD3, TDWRD4, TDLMTS, WRTTD, TDBCNT, RTYCNT
11     ;
12     ;OUTPUTS ARE: DESTRK, RTYCNT, FUNCTN, BUSADR, ERRFNC, RPOF, NEGWRD, ERRMSK
13     ;*
14     ;CALL
15     ;      JSR      PC,WRTTD      ;CALL WRITE TD SUBROUTINE
16     ;      -----      ;RETURN HERE IF EXCEEDED RETRY LIMIT
17     ;      -----      ;ELSE RETURN HERE, NO ERROR
18     ;*
19     ;THIS MODULE IS MAINLY CALLED BY: INPUTD, FETWO, MAIN
20     ;*****
21
22     WRITTD:
23     021152 010046      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
24     021154 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
25     021156 013746 002314  MOV      DESTRK,-(SP) ;;PUSH DESTRK ON STACK
26     23 021162 012700 002352 1#:  MOV      #TDLMTS,R0   ;GET POINTER TO TD LIMITS
27     24 021166 012701 031420  MOV      #TDWRD3,R1   ;GET TRACK DESCRIPTOR POINTER
28     25 021172 112737 000377 002314  MOVVB   #377,DESTRK   ;LOAD THE SECTOR ADDRESS BITS
29     26 021200 021127 140000      CMP      (R1),#NULL   ;ANY DEFECT SKIPS ON THIS TRACK ?
30     27 021204 001414      BEQ      2#          ;BR IF NO
31     28 021206 021120      CMP      (R1),(R0)+   ;WAS TD MOVED ?
32     29 021210 003455      BLE      5#          ;BR IF YES
33     30 021212 032737 040000 031414  BIT      #BIT14,TDWRD1 ;TD BIT ALREADY SET ?
34     31 021220 001051      BNE      5#          ;BR IF YES
35     32 021222 021120      CMP      (R1),(R0)+   ;WAS HDR 0 MOVED ?
36     33 021224 003443      BLE      4#          ;BR IF YES
37     34 021226 032737 100000 031414  BIT      #BIT15,TDWRD1 ;HDR 0 BIT ALREADY SET ?
38     35 021234 001037      BNE      4#          ;BR IF YES
39     36 021236 012737 000065 002404 2#:  MOV      #WRTTD,FUNCTN ;SET UP FOR A WRITE TD FUNCTION
40     37 021244 012737 031414 002234  MOV      #TDWRD1,BUSADR ;GET POINTER TO BEGINNING OF TRACK DESCRIPTOR BUFFER
41     38 021252 012737 005604 002260  MOV      #MSWRTD,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
42     39 021260 052737 100000 002236  BIS      #CMOD,OFFSET ;SET COMMAND MODIFIER BIT
43     40 021266 013737 002330 002376  MOV      TDBCNT,NEGWRD ;6 WORD TRANSFER
44     41 021274 004737 010502      JSR      PC,DRIVER   ;SEND COMMAND TO CONTROLLER
45     42 021300 004737 013304      JSR      PC,RETRY    ;SEE IF WE HAD ANY ERRORS
46     43 021304 000406      BR       3#          ;RETURN HERE IF EXCEEDED RETRY LIMIT
47     44                                ;ELSE RETURN HERE
48     45 021306 005737 002274      TST      RTYCNT      ;DID WE HAVE AN ERROR ?
49     46 021312 001323      BNE      1#          ;BR IF YES
50     47 021314 062766 000002 000006  ADD      #2,6(SP)     ;ADJUST FOR GOOD RETURN
51     48 021322                                3#:  MOV      (SP)+,DESTRK ;;POP STACK INTO DESTRK
52     49 021322 012637 002314      MOV      (SP)+,R1    ;;POP STACK INTO R1
53     50 021326 012601      MOV      (SP)+,R0    ;;POP STACK INTO R0
54     49 021332 000207      RTS      PC          ;RETURN
55     50
56     51 021334 052737 100000 031414 4#:  BIS      #BIT15,TDWRD1 ;MARK HDR 0 MOVED

```


52	021342	000735			BR	2#	;EXECUTE FUNCTION NOW
53							
54	021344	052737	040000	002236	5#:	BIS	#MTD,OFFSET
55	021352	052737	040000	031414		BIS	#BIT14,TDWRD1
56	021360	026127	000002	140000		CMP	2(R1),#NULL
57	021366	001723				BEQ	2#
58	021370	032737	100000	031414		BIT	#BIT15,TDWRD1
59	021376	001356				BNE	4#
60	021400	016146	000002			MOV	2(R1),-(SP)
61	021404	016046	177776			MOV	-2(R0),-(SP)
62	021410	161116				SUB	(R1),(SP)
63	021412	162616				SUB	(SP)+,(SP)
64	021414	016001	000002			MOV	2(R0),R1
65	021420	022601				CMP	(SP)+,R1
66	021422	003305				BGT	2#
67	021424	005746				TST	-(SP)
68	021426	162701	000066			SUB	#<DS/2>,R1
69							
70	021432	022601				CMP	(SP)+,R1
71	021434	003700				BLE	2#
72	021436	000736				BR	4#

```

;MARK TD MOVED IN OFFSET REG
;MARK TD MOVED IN TRACK DESCRIPTOR
;ANY MORE DEFECT SKIPS ON THIS TRACK ?
;BR IF NO
;HDR 0 BIT ALREADY SET ?
;BR IF YES
;SAVE NEXT DEFECT VALUE
;GET UPPER LIMIT FOR TD MOVED
;FIND DIFFERENCE AND
;ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
;GET UPPER LIMIT FOR HDR 0 MOVED
;WAS HEADER 0 ALSO MOVED ?
;BR IF NO
;BACKUP TO GET VALUE AGAIN
;SUBTRACT HALF A DEFECT TO GET LOWER LIMIT
;FOR HDR 0 MOVED
;WAS HEADER 0 ALSO MOVED ?
;BR IF NO

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

.SBTTL READ TRACK DESCRIPTOR MODULE
;*****
;THIS MODULE DOES THE SETUP TO READ A TRACK DESCRIPTOR.  RETRY, IF NECESSARY,
;IS DONE FOR ANY DETECTED ERRORS.
;*
;INPUTS ARE:  RDTD, MSRD TD, TDWRD1, TDBCNT, DESTRK
;
;OUTPUTS ARE:  FUNCTN, ERRFNC, ERRMSK, DESTRK, RPOF, BUSADR, NEGWRD, TDCPY1
;*
;CALL
;      JSR      PC,READTD      ;CALL READ TD SUBROUTINE
;      -----      ;RETURN HERE IF EXCEEDED RETRY LIMIT
;      -----      ;ELSE RETURN HERE, NO ERROR
;*
;THIS MODULE IS MAINLY CALLED BY:  INPUTD, TDSCAN, UPDSCR, MAIN
;*****

READTD:
MOV      FUNCTN, -(SP)      ;PUSH FUNCTN ON STACK
MOV      R0, -(SP)        ;PUSH R0 ON STACK
MOV      R1, -(SP)        ;PUSH R1 ON STACK
MOV      @RDTD, FUNCTN     ;ISSUE THE READ TD COMMAND
MOV      @MSRDTD, ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
1$:      MOV      @TDWRD1, R0 ;GET POINTER TO BEGINNING OF TD BUFFER
MOV      TDBCNT, R1       ;GET BUFFER SIZE
2$:      CLR      (R0)+     ;INITIALIZE THE BUFFER
DEC      R1               ;ONE LESS ENTRY TO GO
BGT      2$              ;IF NOT ZERO, DO AGAIN
MOV      DESTRK, -(SP)    ;PUSH DESTRK ON STACK
MOVB     #377, DESTRK     ;SET LOW BYTE OF ADDRESS
BIS      #CMOD, OFFSET   ;SET COMMAND MODIFIER BIT
MOV      @TDWRD1, BUSADR  ;GET POINTER TO BEGINNING OF TRACK DESCRIPTOR BUFFER
MOV      TDBCNT, NEGWRD   ;6 WORD TRANSFER
JSR      PC, DRIVER      ;NOW DO THE COMMAND
MOV      (SP)+, DESTRK   ;POP STACK INTO DESTRK
JSR      PC, RETRY       ;SEE IF WE HAD ANY ERRORS
BR       7$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
;OTHERWISE RETURN HERE
;DID WE HAVE AN ERROR ?
;BR IF YES
;READ WAS GOOD, SO NOW COMPARE EXPECTED
;AND RECIEVED DATA
;GET POINTER TO BEGINNING OF TD BUFFER
;NOW PREPARE TO CHECK THE CORRECTNESS OF THE DATA
;THIS IS 16 BIT MODE, AND A TRACK DESCRIPTOR
;SET REMAINDER OF EXPECTED BITS
;SET UP FOR THE SECOND WORD OF THE TD
;SET LOW BYTE OF ADDRESS
;CYLINDER ADDRESS MATCH ?
;BR IF NO
;TRACK/SECTOR ADDRESS MATCH ?
;BR IF YES
;LOG 'TD ADDRESS INCORRECT ERROR' STATUS
;TRY READ AGAIN
3$:      MOV      @11, ERRMSK
BR       5$
4$:      TST      (R0)+     ;IS IT VALID DATA ?

```

55	021650	001011				BNE	6#		;BR IF YES
56	021652	012737	000012	002272		MOV	#12,ERRMSK		;LOG 'TD DATA INCORRECT ERROR' STATUS
57	021660	005737	002166		5#:	TST	SCANR		;ARE WE DOING DISK SCAN ?
58	021664	001006				BNE	7#		;BR IF YES
59	021666	004737	013402			JSR	PC,ERRORS		;REPORT THE ERROR
60	021672	000403				BR	7#		
61									
62	021674	062766	000002	000006	6#:	ADD	#2,6(SP)		;ADJUST FOR GOOD RETURN
63	021702				7#:				
	021702	012601				MOV	(SP)+,R1		::POP STACK INTO R1
	021704	012600				MOV	(SP)+,R0		::POP STACK INTO R0
	021706	012637	002404			MOV	(SP)+,FUNCTN		::POP STACK INTO FUNCTN
64	021712	000207				RTS	PC		;OK, END SUB AND RETURN TO MAIN

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47

.SBTTL LIST TD(S) MODULE

```
;;*****  
;THIS MODULE IS USED TO LIST THE TD(S) THAT HAVE DEFECTS FOUND DURING  
;THE LIST OPERATION.  
;*  
;INPUTS ARE: FROM A BUFFER CALLED 'TDWRD1'  
;  
;OUTPUTS ARE: PRESENTED TO THE USER ON THE CONSOLE PRINTER  
;*  
;CALL  
; JSR PC,LISTDS ;CALL ROUTINE  
;*****
```

LISTDS:

```
MOV FUNCTN,-(SP) ;PUSH FUNCTN ON STACK  
MOV R1,-(SP) ;PUSH R1 ON STACK  
MOV R2,-(SP) ;PUSH R2 ON STACK  
MOV R3,-(SP) ;PUSH R3 ON STACK  
CMP TDWRD3,#NULL ;ARE THERE ANY DEFECTS TO LIST ?  
BEQ 8# ;BR IF NO  
MOV #TDDEF,TEXT ;LOAD TEXT WITH ADDRESS OF MESSAGE  
TST LISHDR ;LIST HEADERS IN DEFECT SECTORS ?  
BEQ 1# ;BR IF NO  
MOV #ANDHDR,TEXT ;LOAD TEXT WITH ADDRESS OF MESSAGE  
1#: JSR PC,TDUMP ;GO DUMP THE TRACK DESCRIPTOR  
  
TST LISHDR ;LIST HEADERS IN DEFECT SECTORS ?  
BEQ 8# ;BR IF NO  
MOV #RDHD,FUNCTN ;LOAD A READ HEADER COMMAND  
JSR PC,MAKTRK ;NOW EXECUTE THE COMMAND  
BR 8# ;RETURN HERE IF EXCEEDED RETRY LIMIT  
;ELSE RETURN HERE  
MOV BUSADR,R1 ;GET POINTER TO BEGINNING OF HEADER BLOCK  
MOV TKWCNT,R3 ;GET WORD COUNT FOR TRACK  
  
2#: CMP (R1)+,(R1)+ ;GET OVER HEADER WORDS 1 AND 2  
CMP (R1),#NULL ;ARE THERE ANY DEFECTS TO LIST IN THIS SECTOR ?  
BNE 3# ;BR IF YES  
CMP (R1)+,(R1)+ ;GET OVER HEADER WORDS 3 AND 4  
CMP (R1)+,(R1)+ ;GET OVER HEADER WORDS 5 AND 6  
BR 7#  
3#: MOV -2(R1),-(SP) ;GET THE SECOND WORD OF THE TD (TRK/SEC ADRS)  
MOV (SP),-(SP) ;SAVE SECOND WORD  
CLRB 3(SP) ;GET RID OF HI BYTE (EXTRACT SECTOR)  
;PRINT TD WORD#2, SEC DATA  
  
MOV (SP)+,-(SP)  
MOV (SP)+,-(SP)  
MOV #FRMT37,-(SP)  
MOV #3,-(SP)  
MOV SP,R0  
TRAP C#PNTF  
ADD #10,SP  
44: MOV #4,R2 ;GET MAXIMUM # OF DEFECTS  
45: CMP (R1)+,#NULL ;IS IT NULL DATA ?  
46: BEQ 5# ;BR IF YES  
;PRINT THE DEFECT DATA
```

48	022104	016146	177776		MOV	-2(R1), -(SP)	
	022110	012746	003640		MOV	#FRMT14, -(SP)	
	022114	012746	000002		MOV	#2, -(SP)	
	022120	010600			MOV	SP, R0	
	022122	104417			TRAP	C#PNTF	
	022124	062706	000006		ADD	#6, SP	
49	022130	000412			BR	6#	: AND GO-ON
50	022132			5#:			: PRINT ' 140000 '
51	022132	016146	177776		MOV	-2(R1), -(SP)	
	022136	012746	003651		MOV	#FRMT15, -(SP)	
	022142	012746	000002		MOV	#2, -(SP)	
	022146	010600			MOV	SP, R0	
	022150	104417			TRAP	C#PNTF	
	022152	062706	000006		ADD	#6, SP	
52	022156	005302		6#:	DEC	R2	: ONE LESS ITERATION TO-GO
53	022160	003346			BGT	4#	: DO UNTIL = 0!
54							: CR-LF
55	022162	012746	002656		MOV	#CRLF, -(SP)	
	022166	012746	000001		MOV	#1, -(SP)	
	022172	010600			MOV	SP, R0	
	022174	104417			TRAP	C#PNTF	
	022176	062706	000004		ADD	#4, SP	
56	022202	162703	000006	7#:	SUB	#6, R3	: DONE ALL SECTORS YET ?
57	022206	003303			BGT	2#	: BR IF NO
58	022210			8#:			
	022210	012603			MOV	(SP)+, R3	: POP STACK INTO R3
	022212	012602			MOV	(SP)+, R2	: POP STACK INTO R2
	022214	012601			MOV	(SP)+, R1	: POP STACK INTO R1
	022216	012637	002404		MOV	(SP)+, FUNCTN	: POP STACK INTO FUNCTN
59	022222	000207			RTS	PC	

```

1      .SBTTL  NEW DEFECT LIST DUMP MODULE
2
3      ;;*****
4      ;THIS MODULE IS USED TO PRINT ANY NEW DEFECT DATA FOUND DURING A SCAN
5      ;OPERATION.
6      ;*
7      ;INPUTS ARE:   FROM A BUFFER POINTED BY R1
8      ;
9      ;OUTPUTS ARE:  PRESENTED TO THE USER ON THE CONSOLE PRINTER
10     ;*
11     ;CALL
12     ;      JSR      PC,NEWLST
13     ;*****
14
15     022224  010103
16     022226  105237  002243
17     022232  005046
18     022234  113716  002243
19
20
21     022240  012646
22     022242  012746  004253
23     022246  012746  000002
24     022252  010600
25     022254  104417
26     022256  062706  000006
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41     022374  012646
42     022376  012646
    
```

```

NEWLST: MOV      R1,R3          ;GET THE BUFFER ADDRESS
        INCB     SCANIT        ;INCREMENT SCAN ITERATION COUNT
        CLR      -(SP)         ;SETUP SCAN ITERATION NUMBER FOR TYPEOUT
        MOV      SCANIT,(SP)   ;GET THE ITERATION COUNT
        ;PRINT 'NEW DEFECT(S) DURING SCAN ITERATION XXX..
        ;      TD(S) FOLLOW:'
        MOV      (SP)+, -(SP)
        MOV      #FRMT32, -(SP)
        MOV      #2, -(SP)
        MOV      SP,R0
        TRAP     C#PNTF
        ADD      #6,SP
        ;PRINT 'WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3
        ;      WORD#4 WORD#5 WORD#6'
        MOV      #FRMT35, -(SP)
        MOV      #1, -(SP)
        MOV      SP,R0
        TRAP     C#PNTF
        ADD      #4,SP
14:     TST      (R3)           ;ANY MORE NEW DEFECTS ?
        BEQ      54            ;BR IF NO
        ADD      #<6*2>,R3     ;MOVE TO THE END OF THIS NEW DEFECT TD LIST
        BIT      #BIT8,(R3)+   ;WAS DEFECT BIT SET IN STATUS ?
        BEQ      14           ;BR IF NO
        TST      -(R3)         ;BACKUP R3 POINTER AND
        MOV      R3, -(SP)     ;;PUSH R3 ON STACK
        MOV      #-1, -(SP)    ;;PUSH #-1 ON STACK
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      (SP), -(SP)   ;SAVE 2ND TD WORD
        CLRB     3(SP)         ;GET RID OF HI BYTE (EXTRACT SECTOR)
        MOV      (SP), -(SP)   ;SAVE 2ND TD WORD
        CLRB     2(SP)         ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
        SWAB     2(SP)         ;MAKE IT THE LOW BYTE OF DATA.
        MOV      -(R3), -(SP)  ;;PUSH -(R3) ON STACK
        MOV      (SP), -(SP)   ;SAVE 1ST TD WORD
        BIC      #170000,2(SP) ;GET RID OF THE STATUS BITS
        ;PRINT TD WORD#1, CYL, TD WORD#2 & TRK DATA
        MOV      (SP)+, -(SP)
        MOV      (SP)+, -(SP)
    
```

```

022400 012646          MOV      (SP)+,-(SP)
022402 012646          MOV      (SP)+,-(SP)
022404 012646          MOV      (SP)+,-(SP)
022406 012746 004515   MOV      #FRMT36,-(SP)
022412 012746 000006   MOV      #6,-(SP)
022416 010600          MOV      SP,R0
022420 104417          TRAP     C$PNTF
022422 062706 000016   ADD      #16,SP
42 022426 021627 140000 2$:  CMP      (SP),#NULL      ;IS IT NULL DATA ?
43 022432 001412          BEQ      3$              ;IF MATCH, YES
44                                     ;PRINT THE DATA
45 022434 012646          MOV      (SP)+,-(SP)
022436 012746 003640   MOV      #FRMT14,-(SP)
022442 012746 000002   MOV      #2,-(SP)
022446 010600          MOV      SP,R0
022450 104417          TRAP     C$PNTF
022452 062706 000006   ADD      #6,SP
46 022456 000411          BR       4$              ;AND GO-ON
47 022460          3$:  ;PRINT ' 140000 '
48 022460 012646          MOV      (SP)+,-(SP)
022462 012746 003651   MOV      #FRMT15,-(SP)
022466 012746 000002   MOV      #2,-(SP)
022472 010600          MOV      SP,R0
022474 104417          TRAP     C$PNTF
022476 062706 000006   ADD      #6,SP
49 022502 021627 177777 4$:  CMP      (SP),#-1      ;IS THIS TERMINATOR ?
50 022506 001347          BNE     2$              ;BR IF NO
51 022510 005725          TST     (SP)+          ;RESTORE STACK
52                                     ;CR-LF
53 022512 012746 002656   MOV      #CRLF,-(SP)
022516 012746 000001   MOV      #1,-(SP)
022522 010600          MOV      SP,R0
022524 104417          TRAP     C$PNTF
022526 062706 000004   ADD      #4,SP
54 022532 012603          MOV      (SP)+,R3      ;POP STACK INTO R3
55 022534 005723          TST     (R3)+          ;GET OVER INTERNAL STATUS WORD IN TD LIST
56 022536 020327 120072  CMP      R3,#ENDBUF    ;AT END OF BUFFER?
57 022542 101657          BLOS   1$              ;BR IF NO
58 022544 000207          5$:  RTS      PC

```

```

1      .SBTTL  TD SCAN MODULE
2
3      ;*****
4      ;THIS MODULE IS USED BY THE SCANNER TO DETERMINE IF A TRACK DESCRIPTOR IS
5      ;CORRUPTED.
6      ;*
7      ;OUTPUTS ARE:  TO (R1)+, WHICH IS A BUFFER CALLED 'DEFBUF' AND TO
8      ;                NEWCNT AND ERRCTL
9      ;*
10     ;CALL
11     ;        JSR    PC,TDSCAN
12     ;*****
13
14 022546 012737 000012 002264 TDSCAN: MOV    #10.,ITCNT    ;INITIALIZE ITERATION COUNT
15 022554 013702 002270          MOV    ERRCTL,R2    ;SAVE ERROR TOTAL AT THIS POINT
16 022560 004737 021440 1$:    JSR    PC,READTD    ;CALL THE READ TD SUBROUTINE
17 022564 000406          BR     2$          ;RETURN HERE IF EXCEEDED RETRY LIMIT
18                                     ;ELSE RETURN HERE, NO ERROR
19 022566 023702 002270          CMP    ERRCTL,R2    ;DID WE HAVE AN ERROR BEFORE ?
20 022572 001403          BEQ    2$          ;BR IF NO
21 022574 005737 002264          TST    ITCNT        ;DONE WITH ITERATIONS YET ?
22 022600 003367          BGT    1$          ;BR IF NO
23 022602 163702 002270 2$:    SUB    ERRCTL,R2    ;GET # OF ERRORS DETECTED DURING READ TD
24 022606 005402          NEG    R2          ;MAKE NUMBER POSITIVE AND
25 022610 020227 000002          CMP    R2,#2      ;DID WE HAVE AT LEAST 2 ERRORS ?
26 022614 002003          BGE    3$          ;BR IF YES
27 022616 062716 000002          ADD    #2,(SP)    ;RETURN WITH NO ERROR (TD OK)
28 022622 000422          BR     4$
29
30 022624 013711 002312 3$:    MOV    DESCYL,(R1)  ;START FORMING THE BUFFER NOW
31 022630 052721 030000          BIS    #BIT13!BIT12,(R1)+ ;THIS IS THE FIRST TD WORD.
32 022634 013711 002314          MOV    DESTRK,(R1) ;FORM THE SECOND WORD
33 022640 052721 000377          BIS    #377,(R1)+ ;OF THE TD
34 022644 032761 040000 000010 BIT    #BIT14,10(R1) ;DID WE TRY TO MOVE THE TD ALREADY ?
35 022652 001412          BEQ    5$          ;BR IF NO
36 022654 012737 000001 002272 MOV    #1,ERRMSK    ;LOG 'TD NOT READABLE THRU RELOCATION ERROR' STATUS
37 022662 024141          CMP    -(R1),-(R1) ;BACK UP THE POINTERS
38 022664 005037 002252          CLR    SUPRSS     ;ALLOW ERROR MESSAGE TO BE OUTPUT
39 022670 042761 000400 000014 4$: BIC    #BIT8,14(R1) ;CLEAR INTERNAL DEFECT STATUS AND
40 022676 000207          RTS    PC         ;TAKE HARD ERROR OR NO ERROR RETURN.
41
42 022700 105237 002246 5$:    INCB   NEWCNT      ;INCREMENT THE NEW DEFECT COUNT
43 022704 012721 000166          MOV    #<G1+<DS/2>>,(R1)+ ;GET THE CONSTANT FOR TD MOVED, WORD #3
44 022710 012721 140000          MOV    #NULL,(R1)+   ;NULL TD WORD #4,
45 022714 012721 140000          MOV    #NULL,(R1)+   ;TD WORD #5,
46 022720 012721 140000          MOV    #NULL,(R1)+   ;AND TD WORD #6.
47 022724 052721 040400          BIS    #BIT14!BIT8,(R1)+ ;MARK TD MOVED AND DEFECT IN THE
48 022730 000207          RTS    PC         ;INTERNAL STATUS.

```


4
5
6
7
8
9
10
11
12
13
14
15
16
17
18 024040 013746 002516
19 024044 062716 031000
20 024050 005316
21 024052 005737 002432
22 024056 001407
23 024060 162716 000002
24 024064 032737 004000 002566
25 024072 001401
26 024074 005316
27 024076 162716 000400
28 024102 002375
29 024104 062716 000400
30 024110 012637 002400
31 024114 000207
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47 024116 005037 002402
48 024122 113737 002522 002402
49 024130 001004
50 024132 113737 002322 002402
51 024140 000402
52
53 024142 005337 002402
54 024146 000207

```

;*****
;THIS MODULE IS USED BY 'DASCAN' TO IDENTIFY AND LOCATE ANY DEFECT INFORMATION
;FOUND DURING A SCAN OPERATION.  THE MODULE DETERMINES THE SECTOR ADDRESS OF
;THE NEW DEFECT AND THE WORD COUNT WITHIN THE SECTOR OF THE NEW DEFECT.
;*
;INPUTS ARE:  REG+2 (RPWC)
;
;OUTPUTS ARE: RELWRD (WORD COUNT IN SECTOR)
;*
;CALL
;      JSR      PC,FINDWRD      ;CALL THE ROUTINE
;*****
FINDWRD: MOV     REG+2,-(SP)      ;PUT RPWC CONTENTS ON THE STACK
          ADD     @<256.*50.>,(SP) ;AND CALCULATE THE # OF WORDS XFERD
          DEC     (SP)          ;SUBTRACT 1 FOR AN ODD OR EVEN WORD
          TST     RHTYPE        ;WAS THE CONTROLLER AN RM70 ?
          BEQ     1$           ;BR IF 0, NO
          SUB     @2,(SP)       ;SUBTRACT 2 FOR A DOUBLE WORD
          BIT     @BIT11,REG+52 ;WAS WCE ON AN EVEN WORD?
          BEQ     1$           ;BR IF 0, NO
          DEC     (SP)          ;SUBTRACT 1 FOR AN EVEN WORD
1$:      SUB     @256.,(SP)     ;DID WE FIND THE BAD SECTOR YET ?
          BGE     1$           ;BR IF NO
          ADD     @256.,(SP)     ;THIS IS THE BAD WORD IN THE BAD SECTOR
          MOV     (SP)+,RELWRD   ;SAVE WORD COUNT IN SECTOR
          RTS     PC            ;AND TAKE THE RETURN
;*****
;THIS MODULE IS USED TO FIND THE SECTOR ADDRESS OF THE NEWLY FOUND DEFECT
;WHICH WAS DETECTED DURING THE SCAN OPERATION.
;*
;INPUTS ARE:  REG+6 (RPDA)
;
;OUTPUTS ARE: SECADD (SECTOR IN ERROR)
;*
;CALL
;      JSR      PC,FINDSEC      ;CALL THE ROUTINE
;
;THIS MODULE IS MAINLY CALLED BY: HDSCAN, DASCAN
;*****
FINDSEC: CLR     SECADD         ;INIT SECTOR ADDRESS COUNT
          MOVB    REG+6,SECADD  ;LOAD SECTOR ADDRESS IN ERROR
          BNE     1$           ;BR IF ERROR NOT IN LAST SECTOR, ELSE
          MOVB    SEC50,SECADD  ;SETUP FOR ERROR IN LAST SECTOR
          BR     2$
2$:      DEC     SECADD         ;ADJUST RPDA TO ACCOUNT FOR ERROR
2$:      RTS     PC            ;AND TAKE THE RETURN

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

```

*****
; THIS MODULE IS USED TO DETERMINE IF THE DATA FOUND DURING A SCAN OPERATION
; IS DATA WHICH OCCURS WITH CONSISTENT RESULTS. IF THE DATA BUFFER CONTAINS
; DATA FOR ANY ONE TRACK WHICH DEVIATES MORE THAN +/- 3 WORDS, THE FAILURE IS
; CONSIDERED TO BE A RANDOM WRITE CHECK ERROR, THE DEFECT IS NOT LOGGED IN THE
; TRACK DESCRIPTOR AND THE USER IS NOTIFIED OF THE PROBLEM.
;*
; INPUTS ARE: FROM A BUFFER CALLED 'TDSBLK'
;
; OUTPUTS ARE: RELWRD (AVERAGE DEFECT), ERRMSK (IF ERROR DETECTED)
;*
; CALL
; JSR PC,TOLER ;CALL ROUTINE
*****
TOLER: MOV #5,R0 ;GET POINTER TO HI VALUE FOR WCE'S
CMP #TDSBLK,R4 ;IS THIS THE FIRST WCE DETECTED ?
BNE 1$ ;BR IF NO
MOV (R4),(R0) ;SAVE THIS DEFECT AS HI AND
MOV (R4),2(R0) ;LO VALUES.
1$: CMP (R4),(R0) ;IS THIS VALUE HIGHER THAN PREVIOUS HI ?
BLT 2$ ;BR IF NO
MOV (R4),(R0) ;SAVE THIS AS HI VALUE NOW
BR 3$
2$: TST (R0)+ ;SET POINTER TO LO VALUE
CMP (R4),(R0) ;IS THIS VALUE LOWER THAN PREVIOUS LO ?
BGT 3$ ;BR IF NO
MOV (R4),(R0) ;SAVE THIS AS LO VALUE NOW
TST -(R0) ;SET POINTER TO HI VALUE
3$: MOV (R0)+,-(SP) ;GET HI VALUE AND
MOV (SP),-(SP) ;SAVE IT.
ADD (R0),(SP) ;TOTAL THE HI AND LO VALUES AND
CLC ;CLEAR CARRY BIT
ROR (SP) ;DETERMINE THE AVERAGE.
MOV (SP)+,RELWRD ;SAVE AVERAGE VALUE AS DEFECTIVE SPOT
SUB (R0),(SP) ;GET DIFFERENCE FROM LO TO HI VALUES
CMP (SP)+,#6 ;ARE HI/LO VALUES WITHIN +/-3 WORDS ?
BLE 4$ ;BR IF YES
CLR SUPRS ;ALLOW ERROR MESSAGE TO BE OUTPUT
MOV #-1,ERRMSK ;LOG 'RANDOM ERROR' STATUS
4$: TST (R4)+ ;UPDATE WCE TABLE POINTER
RTS PC ;AND RETURN

5$: .WORD 0 ;HI VALUE GOES HERE
6$: .WORD 0 ;LO VALUE GOES HERE

```

```

024150 012700 024256
024154 022704 033614
024160 001003
024162 011410
024164 011460 000002
024170 021410
024172 002402
024174 011410
024176 000405
024200 005720
024202 021410
024204 003002
024206 011410
024210 005740
024212 012046
024214 011646
024216 061016
024220 000241
024222 006016
024224 012637 002400
024230 161016
024232 022627 000006
024236 003405
024240 005037 002252
024244 012737 177777 002272
024252 005724
024254 000207
024256 000000
024260 000000

```

```

1
2
3 ;*****
4 ;THIS MODULE PROVIDES TEST PATTERNS TO BE USED DURING THE SCANNER PROCESS.
5 ;*
6 ;INPUTS ARE: FROM BGNPAT - ENDPAT (BEGIN PATTERN THRU END PATTERN)
7 ;
8 ;OUTPUTS ARE: TO A BUFFER CALLED 'DBUFF'
9 ;*
10 ;CALL
11 ; JSR PC,GETPAT
12 ;*****
13 GETPAT:
14 024262 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
15 024264 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
16 024266 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
17 024270 013703 002326 MOV PATRN,R3 ;GET THE NEW PATTERN
18 024274 012702 031000 MOV #<256.*50.>,R2 ;GET A WORD/TRACK COUNT
19 024300 012701 036074 MOV #DBUFF,R1 ;GET THE READ/WRITE BUFFER
20 024304 012321 14: MOV (R3)+,(R1)+ ;WRITE FIRST WORD OF PATTERN
21 024306 011321 MOV (R3),(R1)+ ;NOW WRITE THE SECOND WORD
22 024310 005743 TST -(R3) ;BACK UP THE INPUT POINTER
23 024312 162702 000002 SUB #2,R2 ;REDUCE THE ITERATION COUNT
24 024316 003372 BGT 14 ;IF > 0, KEEP GOING
25 024320 022323 CMP (R3)+,(R3)+ ;UPDATE TO NEXT 2 WORD PATTERN
26 024322 010337 002326 MOV R3,PATRN ;STORE THE NEW PATTERN
27 024326 012603 MOV (SP)+,R3 ;POP STACK INTO R3
28 024330 012602 MOV (SP)+,R2 ;POP STACK INTO R2
29 024332 012601 MOV (SP)+,R1 ;POP STACK INTO R1
30 024334 000207 RTS PC ;NOW RETURN TO CALLER

```


1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28

```

*****
THIS MODULE CREATES ABSOLUTE DEFECT DATA, RELATIVE TO INDEX.
*
INPUTS ARE:    FROM A BUFFER CALLED 'TDWRD1'
*
OUTPUTS ARE:   RELWRD
*
CALL
*
JSR    PC,FINDBAD
*****
  
```

```

024474
024474 010146
024476 010246
14 024500 012701 031420
15 024504 012702 000004
16 024510 062137 002400
17 024514 005302
18 024516 003374
19 024520 012701 000155
20 024524 062701 000517
21 024530 005037 002402
22 024534 020137 002400
23 024540 103006
24 024542 005237 002402
25 024546 162737 000517 002400
26 024554 100372
27 024556
   024556 012602
   024560 012601
28 024562 000207
  
```

```

FINDBAD:
MOV    R1,-(SP)    ;;PUSH R1 ON STACK
MOV    R2,-(SP)    ;;PUSH R2 ON STACK
MOV    @TDWRD3,R1  ;;GET THE FIRST DEFECT
MOV    @4,R2        ;;AND THE RECORD SIZE
1$:   ADD    (R1)+,RELWRD ;;ADD THE LIST OF DEFECT TO THE NEW DEFECT
      DEC    R2        ;;ONE LESS RECORD ENTRY
      BGT    1$        ;;OF >0, KEEP GOING
      MOV    @<G1+TD+G2>,R1 ;;THE # OF WORDS BEFORE HDR 0
      ADD    @<HDR+G3+DATA+G4>,R1 ;;ADD THE OVERHEAD FOR A SECTOR
      CLR    SECADD    ;;DEFAULT=SECTOR 0
      CMP    R1,RELWRD  ;;NOW SCALE THE VALUES
      BHS    3$        ;;IF R1 IS GREATER, SECTOR 0 IS BAD
      INC    SECADD    ;;NEXT SECTOR ADDRESS
      SUB    @<HDR+G3+DATA+G4>,RELWRD ;;REDUCE THE REMAINING WORD COUNT
      BPL    2$        ;;OF <0, WE'VE FOUND THE BAD SECTOR
      2$:
3$:   MOV    (SP)+,R2    ;;POP STACK INTO R2
      MOV    (SP)+,R1    ;;POP STACK INTO R1
      RTS    PC        ;;RETURN TO CALLER
  
```

```

1
2
3      ;*****
4      ;THIS MODULE IS USED TO DETERMINE THE HIGHEST VALUE OF A SCRATCH FOUND WITHIN
5      ;THE NEW DEFECT BUFFER.
6      ;*
7      ;OUTPUTS ARE:  HICYL
8      ;*
9      ;CALL
10     ;      JSR      PC,MAXVAL
11     ;*****
12     MAXVAL:
13     024564 010146      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
14     024566 062701 000016  ADD      #<7*2>,R1      ;GET THE NEXT VALUE PAST THE GAP VALUE
15     024572 011146      1$:      MOV      (R1),-(SP)      ;;PUSH (R1) ON STACK
16     024574 042716 170000  BIC      #170000,(SP)    ;STRIP THE UNWANTED BITS
17     024600 022637 002372  CMP      (SP)+,HICYL     ;MATCH?
18     024604 001406      BEQ      3$              ;IF SO, TAKE BRANCH
19     024606 062701 000016  2$:      ADD      #<7*2>,R1    ;GET NEXT BUFFER ITEM
20     024612 020127 036072  CMP      R1,#ENDEDEF    ;OUT OF DEFECT BUFFER YET ?
21     024616 101765      BLOS    1$              ;BR IF NO
22     024620 000414      BR       4$
23     024622 126137 000003 002315  3$:      CMPB    3(R1),DESTRK+1  ;DO WE HAVE A TRACK ADDRESS MATCH?
24     024630 001366      BNE      2$              ;IF NOT, TAKE THE BRANCH
25     024632 004737 024424  JSR      PC,CHEKTD      ;AND CHECK THE EXISTING TRACK DESCRIPTOR VALUES
26     024636 005737 002370  TST      TEMP2          ;CHECK TEMP2 TO DETECT A MATCH
27     024642 001761      BEQ      2$              ;IF ZERO, NO MATCH
28     024644 005237 002372  INC      HICYL          ;INCREASE THE SCRATCH'S HIGHEST CYLINDER
29     024650 000756      BR       2$              ;AND LOOP
30     024652 062737 000005 002372  4$:      ADD      #5,HICYL      ;ADD ANOTHER 5 TO THE SCRATCH'S HIGHEST CYLINDER
31     024660 023737 002372 002212  CMP      HICYL,MAXCYL   ;LEGAL ADDRESS?
32     024666 101403      BLOS    5$              ;IF LOWER OR SAME, YES
33     024670 013737 002212 002372  MOV      MAXCYL,HICYL   ;SET UP FOR ONLY THE MAX CYLINDER
34     024676 012601      5$:      MOV      (SP)+,R1      ;;POP STACK INTO R1
35     024700 000207      RTS      PC              ;RETURN
  
```

```

1
2
3          ;:*****
4          ;THIS MODULE LOOKS FOR GAPS IN THE SCRATCH DATA CONTAINED WITHIN THE NEW
5          ;DEFECT BUFFER.
6          ;*
7          ;OUTPUTS ARE:  'HICYL' (IF A GAP IS FOUND WITHIN 5 CYLINDERS OF THE VALUE
8          ;                      SET BY 'HIVALU' OUTPUT)
9          ;*
10         ;CALL
11         ;          JSR      PC,GAPVAL
12         ;:*****
13 024702          GAPVAL:
14 024702          MOV      R1,-(SP)          ;;PUSH R1 ON STACK
15 024704          MOV      R2,-(SP)          ;;PUSH R2 ON STACK
16 024706          1#:      MOV      (R1),-(SP)          ;;PUSH (R1) ON STACK
17 024710          BIC      @170000,(SP)      ;STRIP THE UNWANTED BITS
18 024714          CMP      (SP)+,HICYL      ;MATCH?
19 024720          BEQ      2#              ;IF EQUAL, THEY MATCH
20 024722          ADD      @<7*2>,R1      ;GET NEXT BUFFER ENTRY
21 024726          CMP      R1,@ENDDF      ;OUT OF DEFECT BUFFER YET ?
22 024732          BLOS     1#              ;BR IF NO
23 024734          BR      6#              ;DONE, IF OVERRUN
24 024736          2#:      CMPB     3(R1),DESTRK+1  ;IS THIS THE CORRECT TRACK ADDRESS?
25 024744          BEQ      3#              ;IF =, YES
26 024746          ADD      @<7*2>,R1      ;GET NEXT BUFFER ENTRY
27 024752          CMP      R1,@ENDDF      ;OUT OF DEFECT BUFFER YET ?
28 024756          BLOS     2#              ;BR IF NO
29 024760          BHI      6#              ;DONE IF OVERRUN
30 024762          3#:      MOV      @5,R2      ;GET THE DELTA VALUE
31 024766          MOV      (R1),DESCYL     ;SAVE THE CURRENT CYLINDER ADDRESS
32 024772          INC      DESCYL         ;AND LOOK FOR THE NEXT VALUE
33 024776          4#:      ADD      @<7*2>,R1      ;GET NEXT BUFFER VALUE
34 025002          CMP      R1,@ENDDF      ;OUT OF DEFECT BUFFER YET ?
35 025006          BHI      6#              ;BR IF YES
36 025010          CMP      (R1),DESCYL     ;CYLINDER ADDRESS MATCH?
37 025014          BLO      4#              ;TAKE BRANCH IF NOT
38 025016          CMPB     3(R1),DESTRK+1  ;TRACK ADDRESS MATCH?
39 025024          BEQ      5#              ;IF SO TAKE BRANCH
40 025026          INC      DESCYL         ;NEXT CYL ADDRESS-PLEASE
41 025032          DEC      R2              ;ONE LESS ITERATION TO GO
42 025034          BGT      4#              ;IF >0, WE'RE NOT DONE
43 025036          BR      6#              ;IF 0 WE NEVER FOUND A CONTINUATION
44 025040          5#:      JSR      PC,CHEKTD    ;GET THE NEW TD SUMMATION
45 025044          TST      TEMP2         ;IF A SCRATCH VALUE FOUND, THIS <>0
46 025050          BEQ      6#              ;TAKE BRANCH IF SCRATCH VALUE NOT FOUND
47 025052          MOV      @5,-(SP)      ;GET THE MAX VALUE FOR INTERPOLATION
48 025056          SUB      R2,(SP)      ;AND DERIVE THE IMPLIED SCRATCH WIDTH
49 025060          ADD      (SP)+,HICYL     ;ADD THAT WIDTH TO THE HIGH LIMIT
50 025064          6#:      MOV      (SP)+,R2      ;;POP STACK INTO R2
51 025066          MOV      (SP)+,R1      ;;POP STACK INTO R1
52 025070          RTS      PC             ;RETURN TO CALLER

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13 025072
14 025072 010146
15 025074 062701 000016
16 025100 126137 000003 002315
17 025106 001007
18 025110 004737 024424
19 025114 005737 002370
20 025120 001402
21 025122 005237 002372
22 025126 062701 000016
23 025132 020127 036072
24 025136 101760
25 025140 012601
26 025142 000207
27
28
29
30
31
32
33
34
35
36
37 025144
38 025144 010146
39 025146 010246
40 025150 010102
41 025152 062702 000016
42 025156 012221
43 025160 020227 036072
44 025164 101774
45 025166 012602
46 025170 012601
47 025172 000207

;*****
;THIS MODULE IS USED AS PART OF THE SCRATCH INTERPOLATION PROCESS IT'S PURPOSE
;IS TO DETERMINE THE HIGHEST CONTINUOUS ADDRESS WHERE A SCRATCH HAS BEEN
;LOCATED.
;*
;OUTPUTS ARE:  HICYL
;*
;CALL
;        JSR    PC,HIVALU
;*****

HIVALU:
        MOV     R1,-(SP)           ;;PUSH R1 ON STACK
        ADD     @<7*2>,R1        ;;MOVE INPUT OVER THE FIRST DEFECT
1$:     CMPB    3(R1),DESTRK+1    ;;TRACK MATCH ?
        BNE     2$                ;;IF NOT, TAKE BRANCH
        JSR     PC,CHEKTD        ;;DOES THE TD DATA MATCH?
        TST     TEMP2           ;;IF TEMP2=0,NO!
        BEQ     2$                ;;TAKE BRANCH IF NO MATCH
        INC     HICYL           ;;INCLUDE THIS CYLINDER IN THE SCRATCH AREA
2$:     ADD     @<7*2>,R1        ;;NEXT ENTRY
        CMP     R1,@ENDEF       ;;OUT OF DEFECT BUFFER YET ?
        BLOS   1$                ;;BR IF NO
        MOV     (SP)+,R1        ;;POP STACK INTO R1
        RTS     PC                ;;RETURN

;*****
;THIS MODULE IS USED TO COMPRESS DATA OUT OF THE BUFFER, THAT DATA WHICH HAS
;BEEN INTERPOLATED AS NON-CONTIGUOUS SCRATCH DATA. THE DATA WHICH REMAINS IS
;DATA REPRESENTATIVE OF DEFECTS ACTUALLY FOUND ON THE MEDIA, NOT INTERPOLATED
;DATA.
;*
;CALL
;        JSR    PC,COMPRS
;*****

COMPRS:
        MOV     R1,-(SP)           ;;PUSH R1 ON STACK
        MOV     R2,-(SP)           ;;PUSH R2 ON STACK
        MOV     R1,R2             ;;GET THE CURRENT BUFFER ADDRESS
1$:     ADD     @<7*2>,R2        ;;GET THE NEXT ENTRY
        MOV     (R2)+,(R1)+       ;;COMPRESS OVER THE CURRENT ENTRY
        CMP     R2,@ENDEF       ;;OUT OF DEFECT BUFFER YET ?
        BLOS   1$                ;;BR IF NO
        MOV     (SP)+,R2        ;;POP STACK INTO R2
        MOV     (SP)+,R1        ;;POP STACK INTO R1
        RTS     PC

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

;*****
;THIS MODULE OUTPUTS DATA TO THE NEW DEFECT BUFFER FOR EVENTUAL TO REWRITE OR
;IT PRODUCES OUTPUT TO THE BAD SECTOR FILE WHEN A TD IS FOUND TO CONTAIN 4
;DEFECTS.
;*
;OUTPUTS ARE:  TO (R2)+, A BUFFER CALLED 'MFGBUF' OR TO (R4)+, A BUFFER
;              CALLED 'DBUFF'
;*
;CALL
;              JSR     PC,UPDSCR
;*****

UPDSCR:
      MOV     R1,-(SP)           ;;PUSH R1 ON STACK
      MOV     R3,-(SP)           ;;PUSH R3 ON STACK
      JSR     PC,READTD          ;CALL THE READ TD SUBROUTINE
      BR      1$                ;RETURN HERE IF EXCEEDED RETRY LIMIT
      BR      2$                ;ELSE RETURN HERE, NO ERROR
1$:   JSR     PC,DESTD           ;GET DESIRED TD, BECAUSE WE FAILED THE READ
2$:   MOV     RELWRD,-(SP)        ;PUSH RELWRD ON STACK
      JSR     PC,INSERT          ;INSERT THE DEFECT IN TD
      MOV     (SP)+,RELWRD       ;POP STACK INTO RELWRD
      TST     TMPBSF             ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
      BMI     3$                ;BR IF YES
      JSR     PC,FINDBAD          ;FIND THE BAD SECTOR
      MOVB    SECADD,TMPBSF+2    ;SAVE THE BAD SECTOR AND
      JSR     PC,LODBSF          ;LOAD THE BSF BUFFER
      BR      5$

3$:   INCB    NEWCNT             ;ADD ONE TO THE TOTAL DEFECT COUNT
      MOV     #6,R3              ;GET # OF ENTRIES IN TRACK DESCRIPTOR
      MOV     #TDWRD1,R1        ;GET POINTER TO BEGINNING OF TD BUFFER
4$:   MOV     (R1)+,(R4)+        ;PUT THE NEW DEFECT INTO BUFFER
      DEC     R3                 ;DONE ALL ENTRIES YET ?
      BGT     4$                ;BR IF NO
      BIS     #BIT9,(R4)+       ;MARK A SCRATCH, IN THE INTERNAL STATUS
5$:   MOV     (SP)+,R3           ;;POP STACK INTO R3
      MOV     (SP)+,R1           ;;POP STACK INTO R1
      RTS     PC                 ;NOW RETURN
    
```

025174
 025174 010146
 025176 010346
 15 025200 004737 021440
 16 025204 000401
 17 025206 000402
 18 025210 004737 026332
 19 025214
 025214 013746 002400
 20 025220 004737 016616
 21 025224 012637 002400
 22 025230 005737 031460
 23 025234 100410
 24 025236 004737 024474
 25 025242 113737 002402 031462
 26 025250 004737 011476
 27 025254 000413
 28
 29 025256 105237 002246
 30 025262 012703 000006
 31 025266 012701 031414
 32 025272 012124
 33 025274 005303
 34 025276 003375
 35 025300 052724 001000
 36 025304
 025304 012603
 025306 012601
 37 025310 000207


```
112  
113 026026 0E2701 000016 16: ADD @<7*2>,R1 ;NEXT BUFFER ENTRY, PLEASE  
114 026032 026637 177776 002372 17: CMP -2(SP),HICYL ;STILL IN RANGE?  
115 026040 101402 BLOS 18: ;IF SO, TAKE THE BRANCH  
116 026042 000137 025406 JMP 4: ;GET THE NEXT TRACK AND LOOK FOR ANOTHER SCRATCH  
117  
118 026046 126137 000003 002315 18: CMPB 3(R1),DESTRK+1 ;TRACK MATCH ?  
119 026054 001402 BEQ 19: ;IF SO, LOG THE DATA  
120 026056 103763 BLO 16: ;IF LOWER, KEEP MOVING THROUGH THE BUFFER  
121 026060 101346 BHI 14: ;IF HIGHER, GET THE TD FROM THE DISK, AND UPDATE IT  
122 026062 012705 000007 19: MOV @7,R5 ;7 ITEMS / RECORD  
123 026066 012124 20: MOV (R1)+,(R4)+ ;LOG THIS SCRATCH DATA  
124 026070 005305 DEC R5 ;ONE LESS ITEM / THIS RECORD  
125 026072 003375 BGT 20: ;KEEP GOING UNTIL R5 = 0  
126 026074 162701 000016 SUB @<7*2>,R1 ;BACK R1 UP TO THE TOP OF THIS RECORD  
127 026100 004737 025144 JSR PC,COMPRS ;STRIP OUT THIS DEFECT, AND COMPRESS THE BUFFER  
128 026104 005237 002312 INC DESCYL ;NEXT DESIRED CYLINDER PLEASE  
129 026110 005237 002374 INC LOCYL ;ONE LESS CYLINDER TO-GO  
130 026114 020127 036072 CMP R1,#ENDDF ;END OF DEFECT BUFFER YET ?  
131 026120 101734 BLOS 15: ;BR IF NO  
132 026122 000137 025350 JMP 2: ;GET THE NEXT CYLINDER ADDRESS  
133  
134 026126 012701 032464 21: MOV #DEFBUF,R1 ;GET POINTER TO BEGINNING OF DEFECT BUFFER  
135 026132 012705 000007 22: MOV @7,R5 ;SEVEN ENTRIES / RECORD  
136 026136 012124 23: MOV (R1)+,(R4)+ ;LOG THE NON SCRATCH RELATED DATA  
137 026140 005305 DEC R5 ;ONE LESS ENTRY THIS RECORD  
138 026142 003375 BGT 23: ;DO UNTIL R5 = 0  
139 026144 005711 TST (R1) ;IS NEXT ENTRY VALID DATA ?  
140 026146 001371 BNE 22: ;BR IF YES  
141 026150 012637 002314 MOV (SP)+,DESTRK ;POP STACK INTO DESTRK  
026154 012637 002312 MOV (SP)+,DESCYL ;POP STACK INTO DESCYL  
142 026160 000207 RTS PC
```


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53

026162
026162 010046
026164 012700 031414
026170 011137 002312
026174 042737 170000 002312
026202 012120
026204 011137 002314
026210 105037 002314
026214 012120
026216 012120
026220 012120
026222 012120
026224 012110
026226 012600
026230 032721 000400
026234 000207

026236
026236 010046
026240 010146
026242 012700 031414
026246 012701 031444
026252 012021
026254 012021
026256 012021
026260 012021
026262 012021
026264 011011
026266 012601
026270 012600
026272 000207

```

;*****
;THIS MODULE IS USED BY THE SCANNER MODE OF OPERATION
;
;INPUTS ARE: FROM (R1)+,(R1)+,(R1)+,(R1)+,(R1)+,(R1)+,(R1)+
;
;OUTPUTS ARE: DESCYL, DESTRK, TRACK DESCRIPTOR BUFFER CALLED 'TDWRD1',
;              UPDATE DEFECT BUFFER POINTER IN R1 AND TEST THE INTERNAL
;              DEFECT STATUS BEFORE RETURNING.
;
;THIS MODULE IS CALLED BY: MAIN
;*****
    
```

```

LOADTD:
    MOV     RO,-(SP)           ;;PUSH RO ON STACK
    MOV     @TDWRD1,RO        ;;GET POINTER TO BEGINNING OF TD BUFFER
    MOV     (R1),DESCYL       ;;LOAD THE DESIRED CYLINDER ADDRESS AND
    BIC     #170000,DESCYL    ;;STRIP OFF STATUS BITS.
    MOV     (R1)+,(RO)+       ;;LOAD CYLINDER ADDRESS INTO TD WORD #1
    MOV     (R1),DESTRK       ;;LOAD THE TRACK ADDRESS AND
    CLRB    DESTRK           ;;THE SECTOR ADDRESS.
    MOV     (R1)+,(RO)+       ;;LOAD TRACK/SECTOR ADRS INTO TD WORD #2
    MOV     (R1)+,(RO)+       ;;LOAD TD WORD #3,
    MOV     (R1)+,(RO)+       ;;TD WORD #4,
    MOV     (R1)+,(RO)+       ;;TD WORD #5,
    MOV     (R1)+,(RO)        ;;AND TD WORD #6.
    MOV     (SP)+,RO          ;;POP STACK INTO RO
    BIT     @BIT8,(R1)+       ;;GET OVER INTERNAL STATUS WORD AND
    RTS     PC                 ;;TEST DEFECT STATUS BIT IN TD LIST.
    ;;TAKE THE RETURN NOW
    
```

```

;*****
;THIS MODULE IS USED TO SAVE THE CONTENTS OF A BUFFER CALLED 'TDWRD1' IN THE
;A BUFFER CALLED 'TDTMP1'.
;
;INPUTS ARE: TDWRD1 - TDWRD6
;
;OUTPUTS ARE: TDTMP1 - TDTMP6
;
;THIS MODULE IS CALLED BY: MAIN
;*****
    
```

```

SAVETD:
    MOV     RO,-(SP)           ;;PUSH RO ON STACK
    MOV     R1,-(SP)          ;;PUSH R1 ON STACK
    MOV     @TDWRD1,RO        ;;GET POINTER TO BEGINNING OF TD BUFFER
    MOV     @TDTMP1,R1        ;;GET POINTER TO BEGINNING OF TD TEMP. BUFFER
    MOV     (RO)+,(R1)+       ;;SAVE TD WORD #1,
    MOV     (RO)+,(R1)+       ;;TD WORD #2,
    MOV     (RO)+,(R1)+       ;;TD WORD #3,
    MOV     (RO)+,(R1)+       ;;TD WORD #4,
    MOV     (RO)+,(R1)+       ;;TD WORD #5,
    MOV     (RO),(R1)         ;;AND TD WORD #6.
    MOV     (SP)+,R1          ;;POP STACK INTO R1
    MOV     (SP)+,RO          ;;POP STACK INTO RO
    RTS     PC
    
```

```

54
55
56
57
58
59
60
61
62
63
64
65 026274
026274 010046
026276 010146
66 026300 012700 031444
67 026304 012701 031414
68 026310 012021
69 026312 012021
70 026314 012021
71 026316 012021
72 026320 012021
73 026322 011011
74 026324 012601
026326 012600
75 026330 000207
76
77
78
79
80
81
82
83
84
85
86
87
88
89 026332
026332 010046
90 026334 012700 031414
91 026340 013710 002312
92 026344 052720 030000
93 026350 013710 002314
94 026354 052720 000377
95 026360 012720 140000
96 026364 012720 140000
97 026370 012720 140000
98 026374 012710 140000
99 026400 012600
100 026402 000207

```

```

;*****
;THIS MODULE IS USED TO RESTORE THE CONTENTS OF A BUFFER CALLED 'TDTMP1' WITH
;THE CONTENTS OF A BUFFER CALLED 'TDWRD1'.
;*
;INPUTS ARE:   TDTMP1 - TDTMP6
;
;OUTPUTS ARE:  TDWRD1 - TDWRD6
;*
;THIS MODULE IS CALLED BY: MAIN
;*****
RESTTD:
MOV     RO,-(SP)           ;;PUSH RO ON STACK
MOV     R1,-(SP)           ;;PUSH R1 ON STACK
MOV     @TDTMP1,R0         ;GET POINTER TO BEGINNING OF TD TEMP. BUFFER
MOV     @TDWRD1,R1        ;GET POINTER TO BEGINNING OF TD BUFFER
MOV     (R0),.(R1)+       ;SAVE TD WORD #1.
MOV     (R0),.(R1)+       ;TD WORD #2.
MOV     (R0),.(R1)+       ;TD WORD #3.
MOV     (R0),.(R1)+       ;TD WORD #4.
MOV     (R0),.(R1)+       ;TD WORD #5.
MOV     (R0),.(R1)        ;AND TD WORD #6.
MOV     (SP),R1           ;;POP STACK INTO R1
MOV     (SP),RO           ;;POP STACK INTO RO
RTS     PC

```

```

;*****
;THIS MODULE IS USED WHENEVER THE TD CANNOT BE READ CORRECTLY AND THE
;DESIRED CYLINDER/TRACK/SECTOR ADDRESSES MUST BE SETUP IN THE TD BUFFER.
;ALSO, THE FOUR DEFECT WORDS OF THE TD ARE NULLED.
;*
;INPUTS ARE:   DESCYL, DESTRK
;
;OUTPUTS ARE:  TO TRACK DESCRIPTOR BUFFER CALLED 'TDWRD1'
;*
;THIS MODULE IS CALLED BY: MAIN
;*****
DESTD:
MOV     RO,-(SP)           ;;PUSH RO ON STACK
MOV     @TDWRD1,R0         ;POINT TO BEGINNING OF TD BUFFER
MOV     DESCYL,(R0)        ;LOAD THE CYLINDER ADDRESS AND
BIS     @BIT13:BIT12,(R0)+  ;MARK WORD AS TD & FMT16 IN TD WORD #1
MOV     DESTRK,(R0)        ;LOAD THE TRACK ADDRESS AND
BIS     @377,(R0)+         ;THE SECTOR ADDRESS IN TD WORD #2
MOV     #NULL,(R0)+       ;NULL TD WORD #3.
MOV     #NULL,(R0)+       ;TD WORD #4.
MOV     #NULL,(R0)+       ;TD WORD #5.
MOV     #NULL,(R0)        ;AND TD WORD #6.
MOV     (SP),RO           ;;POP STACK INTO RO
RTS     PC                 ;TAKE THE RETURN NOW

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
37
44

```

;*****
;THIS MODULE IS USED TO RESTORE THE MASTER BUFFER 'DBUFF' BETWEEN ITERATIONS
;OF THE SCANNER.
;*****
    
```

```

7 026404 012701 032464 RESTOR: MOV #DEFBUF,R1 ;GET POINTER TO BEGINNING OF DEFECT BUFFER
8 026410 012703 036074 MOV #DBUFF,R3 ;GET POINTER TO BEGINNING OF DATA BUFFER
9 026414 005713 1#: TST (R3) ;IS THERE A DEFECT ENTRY HERE ?
10 026416 001430 BEQ 5# ;BR IF NO
11 026420 032763 001000 000014 BIT #BIT9,14(R3) ;WAS DEFECT AN INTERPOLATED SCRATCH ?
12 026426 001004 BNE 2# ;BR IF YES
13 026430 032763 000400 000014 BIT #BIT8,14(R3) ;WAS DEFECT BIT SET IN STATUS ?
14 026436 001006 BNE 3# ;BR IF YES
15 026440 062703 000016 2#: ADD #<7*2>,R3 ;DO NOT RESTORE ENTRY BACK TO DEFECT BUFFER
16 026444 020327 120072 CMP R3,#ENDBUF ;DONE WITH BUFFER YET ?
17 026450 101761 BLOS 1# ;BR IF NO
18 026452 000412 BR 5#
19
20 026454 012704 000007 3#: MOV #7,R4 ;SEVEN ENTRIES / RECORD
21 026460 012321 4#: MOV (R3)+,(R1)+ ;RESTORE THE ORIGINAL BUFFER
22 026462 005304 DEC R4 ;ONE LESS ENTRY THIS RECORD
23 026464 003375 BGT 4# ;DO UNTIL R4 = 0
24 026466 010137 002240 MOV R1,LASLOC ;SAVE THIS THE 1ST UNUSED LOCATION FOR NOW
25 026472 020127 036072 CMP R1,#ENDEFF ;END OF DEFECT BUFFER YET ?
26 026476 101746 BLOS 1# ;BR IF NO
27 026500 000207 5#: RTS PC
    
```

12
40
42
43
44
45
46
47 026502
48
60
61 026502 000167
026504 000000
62
74
75
76 026506
026506 104425

.SBTTL REPORT CODING SECTION
; **
; THE REPORT CODING SECTION CONTAINS THE
; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
; --
L\$RPT::

.WORD J\$JMP
.WORD L10007-2-.

.EVEN

L10007: TRAP C\$RPT

```

1
2
3
4
5
6
7
8 026510
9 026510 000000
10 026512 177777
11 026514 000006
13

```

.SBTTL PROTECTION TABLE

```

; **
; THIS TABLE IS USED BY THE RUNTIME SERVICES
; TO PROTECT THE LOAD MEDIA.
; --

```

```

L$PROT::
      0      ;P-TABLE OFFSET OF CSR
     -1     ;NOT A MASSBUS DEVICE
      6     ;P-TABLE OFFSET DRIVE #

```

```

1          .SBTTL  INITIALIZE SECTION
2
3          ;**
4          ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
5          ; AT THE BEGINNING OF EACH PASS.
6          ;--
7
8 026516    L$INIT::
9
10 026516   104433          TRAP      C$RESET          ;RESET THE WORLD
11
12 026520   012700   000034  MOV      @EF.PWR,RO    ;POWER FAIL START ?
13 026524   104447          TRAP      C$REFG
14 026526   103002          BCC      1$          ;GO TO 1$ IF NO
15
16 026530   104432          TRAP      C$EXIT          ;EXIT IF YES
17 026532   001014          .WORD    L10011-.
18 026534   012700   000036  1$:      MOV      @EF.CON,RO    ;CONTINUE COMMAND ?
19 026540   104447          TRAP      C$REFG
20
21 026542   103004          BCC      2$          ;GO TO 2$ IF NO
22 026544   105237   002247  INCB    CONFLG        ;INDICATE CONTINUE COMMAND AND EXIT
23 026550   104432          TRAP      C$EXIT
24 026552   000774          .WORD    L10011-.
25
26 026554   012700   000037  2$:      MOV      @EF.RES,RO    ;RESTART COMMAND ?
27 026560   104447          TRAP      C$REFG
28
29 026562   103406          BCS      3$          ;GO TO 3$ IF YES
30
31 026564   012700   000040  MOV      @EF.STA,RO    ;START COMMAND ?
32 026570   104447          TRAP      C$REFG
33
34 026572   103402          BCS      3$          ;GO TO 3$ IF YES
35 026574   000137   027260  JMP     15$          ;GO TO 15$ IF NO
36
37 026600   005037   002114  3$:      CLR      L$TEST        ;CLEAR TEST NUMBER
38 026604   005227   177777  4$:      INC      @-1          ;FIRST TIME THRU HERE ?
39 026610   001020          BNE      6$          ;BR IF NO
40 026612          5$:      MOV      @FRMT50,-(SP) ;PRINT THE HELP MESSAGE
41 026616   012746   004634  MOV      @1,-(SP)
42 026622   010600          MOV      SP,RO
43 026624   104417          TRAP      C$PNTF
44 026626   062706   000004  ADD     @4,SP
45
46 026632   012746   004742          ;PRINT THE HELP MESSAGE
47 026636   012746   000001  MOV      @FRMT51,-(SP)
48 026642   010600          MOV      @1,-(SP)
49 026644   104417          MOV      SP,RO
50 026646   062706   000004  TRAP      C$PNTF
51
52 026652          6$:      ADD     @4,SP
53
54 026652   104443          TRAP      C$GMAN
55 026654   000406          BR       10000$
56 026656   002160          .WORD    OPTION

```

```

026660 000052          .WORD  T%CODE
026662 006254          .WORD  MSG7
026664 000007          .WORD  7
026666 000000          .WORD  T%LOLIM
026670 000007          .WORD  T%HILIM
026672          10000%:
43 026672 005037 002156 CLR      ANYWHR      ;DEFAULT; ACCESS TO FE CYLINDER ONLY
44 026676 013737 002214 002212 MOV     TMPCYL,MAXCYL ;RESTORE MAXIMUM CYLINDER VALUE
45 026704 005737 002160 TST     OPTION      ;FORMAT OPTION ?
46 026710 001413 BEQ     7%          ;BR IF YES
47 026712 023727 002160 000002 CMP     OPTION,%2   ;SCAN OPTION ?
48 026720 001407 BEQ     7%          ;BR IF YES
49 026722 023727 002160 000004 CMP     OPTION,%4   ;MODIFY OPTION ?
50 026730 001403 BEQ     7%          ;BR IF YES
51 026732 005237 002156 INC     ANYWHR      ;ALLOW ACCESS TO ANYWHERE ON THE MEDIA
52 026736 000417 BR      8%
53 026740          7%:
54 026740 104443 TRAP   C%GMAN
026742 000404 BR      10001%
026744 002156 .WORD  ANYWHR
026746 000130 .WORD  T%CODE
026750 006342 .WORD  WRITMG
026752 000001 .WORD  1
026754          10001%:
55 026754 005737 002156 TST     ANYWHR      ;SEE IF ACCESS IS TO ANYWHERE ON THE MEDIA.
56 026760 001406 BEQ     8%          ;BRANCH IF NOT, ELSE
57                                     ;TELL USER THAT DATA WILL BE DESTROYED.
58                                     ;PRINT '! CUSTOMER DATA WILL BE OVERWRITTEN !
59                                     ;
60                                     ;-----
61                                     ;CONTINUE ?'
61 026762 104443 TRAP   C%GMAN
026764 000404 BR      10002%
026766 002156 .WORD  ANYWHR
026770 000120 .WORD  T%CODE
026772 006411 .WORD  WRSAFM
026774 000001 .WORD  1
026776          10002%:
63 026776 023727 002160 000006 8%: CMP     OPTION,%6   ;PRINT HELP MESSAGE ?
64 027004 001702 BEQ     5%          ;BR IF YES
65 027006 023727 002160 000004 CMP     OPTION,%4   ;WHAT IS OPTION ?
66 027014 002116 BGE    14%
67                                     ;BRANCH IF MODIFY OR WRITE FE-2
68 027016 005037 002206 CLR     DRVPAR      ;ELSE, OPTION IS FORMAT, VERIFY, SCAN OR LIST
69                                     ;DEFAULT TO 'N' FOR CHANGE PARAMETERS
70                                     ;PRINT 'CHANGE DRIVE PARAMETERS (L) N ?'
70 027022 104443 TRAP   C%GMAN
027024 000404 BR      10003%
027026 002206 .WORD  DRVPAR
027030 000130 .WORD  T%CODE
027032 006224 .WORD  MSG18
027034 000001 .WORD  1
027036          10003%:
71 027036 005737 002206 TST     DRVPAR      ;CHANGE DRIVE PARAMETERS ?
72 027042 001446 BEQ     10%         ;BR IF NO
74 027044 005737 002156 TST     ANYWHR      ;DON'T ALLOW MIN/MAX CYLINDER TO BE CHANGED.
75 027050 001423 BEQ     9%          ;IF ACCESS IS TO FE CYLINDER ONLY, ELSE
77                                     ;PRINT 'MIN CYL (D) 0 ?'
78 027052 104443 TRAP   C%GMAN

```

```

027054 000406          BR      10004$
027056 002210          .WORD  MINCYL
027060 000052          .WORD  T$CODE
027062 006322          .WORD  MMSG15
027064 001777          .WORD  1777
027066 000000          .WORD  T$LOLIM
027070 001166          .WORD  T$HILIM
027072          10004$:
79
80 027072 104443          TRAP  C$GMAN          ;PRINT 'MAX CYL (D) 630 ?'
027074 000406          BR      10005$
027076 002212          .WORD  MAXCYL
027100 000052          .WORD  T$CODE
027102 006332          .WORD  MMSG16
027104 001777          .WORD  1777
027106 000000          .WORD  T$LOLIM
027110 001166          .WORD  T$HILIM
027112          10005$:
82 027112 013737 002212 002214  MOV   MAXCYL, TMPCYL  ;SAVE MAXIMUM CYLINDER VALUE
84 027120          9$:          ;PRINT 'MIN TRK (D) 0 ?'
85 027120 104443          TRAP  C$GMAN
027122 000406          BR      10006$
027124 002216          .WORD  MINTRK
027126 000052          .WORD  T$CODE
027130 006302          .WORD  MMSG13
027132 000037          .WORD  37
027134 000000          .WORD  T$LOLIM
027136 000037          .WORD  T$HILIM
027140          10006$:
86
87 027140 104443          TRAP  C$GMAN          ;PRINT 'MAX TRK (D) 31 ?'
027142 000406          BR      10007$
027144 002220          .WORD  MAXTRK
027146 000052          .WORD  T$CODE
027150 006312          .WORD  MMSG14
027152 000037          .WORD  37
027154 000000          .WORD  T$LOLIM
027156 000037          .WORD  T$HILIM
027160          10007$:
88
89 027160 005737 002160          10$:  TST   OPTION          ;FORMAT OPTION ?
90 027164 001006          BNE   12$          ;BR IF NO
91 027166          11$:
92 027166 104443          TRAP  C$GMAN
027170 000404          BR      10010$
027172 002176          .WORD  NOWRCK
027174 000130          .WORD  T$CODE
027176 006544          .WORD  MMSG17
027200 000001          .WORD  1
027202          10010$:
93
94 027202 023727 002160 000002 12$:  CMP   OPTION, #2    ;SCAN OPTION ?
95 027210 001006          BNE   13$          ;BR IF NO
96
97
98 027212 104443          TRAP  C$GMAN
027214 000404          BR      10011$          ;PRINT 'DO YOU WANT TO RE-WRITE TD(S) WITH
;          NEW DEFECTS ?'

```



```

027216 002200          .WORD  ENWTTD
027220 000130          .WORD  T$CODE
027222 006642          .WORD  MSG19
027224 000001          .WORD  1
027226                                10011$:
99
100 027226 023727 002160 000003 13$:  CMP  OPTION,#3      ;LIST OPTION ?
101 027234 001006                                BNE  14$          ;BR IF NO
102
103 027236 104443          TRAP  C$GMAN
027240 000404          BR    10012$
027242 002204          .WORD  LISHDR
027244 000130          .WORD  T$CODE
027246 006721          .WORD  MSG23
027250 000001          .WORD  1
027252                                10012$:

```

```

1
2
3           ;SETUP UNIT TO BE TESTED FROM P-TABLE AND DETERMINE WHAT OPTION
4           ;SHOULD BE PERFORMED.
5 027252 012737 177777 002420 14$: MOV    #-1,UNIT      ;INITIALIZE FOR UNIT 0 ON START
6 027260 005237 002420          15$: INC    UNIT          ;INCREMENT TO NEXT UNIT
7 027264 004737 016322          JSR    PC,TABELD     ;LOAD THE HARDWARE P-TABLES
8 027270 023737 002420 002012  CMP    UNIT,L$UNIT   ;OUT OF UNITS TO TEST ?
9 027276 002403                BLT    16$           ;BR IF NO
10 027300 104444                TRAP  C$DCLN
11 027302 104432                TRAP  C$EXIT
    027304 000242                .WORD L10011-.
12 027306                16$:           ;SET PRIORITY TO 7
13 027306 012700 000340        MOV    #PRI07,R0
    027312 104441                TRAP  C$SPRI
14
15           ;RESET SOME COMMON TABLES AND PARAMETERS
16
17 027314 010146                MOV    R1,-(SP)      ;;PUSH R1 ON STACK
18 027316 012701 002162        MOV    #FORMT,R1    ;LOAD THE TOP OF THE CONTROL FILE
19 027322 005021                17$: CLR    (R1)+       ;AND RESET IT
20 027324 020127 002174        CMP    R1,#WRTFE2   ;FINISHED?
21 027330 101774                BLOS  17$           ;IF NOT >, NO!
22 027332 012701 002242        MOV    #FIRPAS,R1  ;LOAD THE TOP OF THE CONTROL FILE
23 027336 005021                18$: CLR    (R1)+       ;AND RESET IT
24 027340 020127 002320        CMP    R1,#SAVTRK  ;FINISHED?
25 027344 101774                BLOS  18$           ;IF NOT >, NO!
26 027346 012701 002600        MOV    #CMDQUE,R1  ;SET R1 = THE TOP OF THE COMMAND QUEUE
27
28           ;FIND WHAT OPTION WAS SPECIFIED
29
30 027352 005737 002160        TST   OPTION        ;FORMAT OPTION ?
31 027356 001003                BNE   19$           ;BR IF NO
32 027360 005237 002162        INC   FORMT         ;SET FORMAT OPTION
33 027364 000437                BR    24$
34 027366 023727 002160 000001 19$: CMP    OPTION,#1    ;VERIFY OPTION ?
35 027374 001003                BNE   20$           ;BR IF NO
36 027376 005237 002164        INC   VRIFY         ;SET VERIFY OPTION
37 027402 000433                BR    25$
38 027404 023727 002160 000002 20$: CMP    OPTION,#2    ;SCAN OPTION ?
39 027412 001003                BNE   21$           ;BR IF NO
40 027414 005237 002166        INC   SCANR        ;SET SCAN OPTION
41 027420 000442                BR    29$
42 027422 023727 002160 000003 21$: CMP    OPTION,#3    ;LIST OPTION ?
43 027430 001003                BNE   22$           ;BR IF NO
44 027432 005237 002172        INC   LIST         ;SET LIST OPTION
45 027436 000431                BR    28$
46 027440 023727 002160 000004 22$: CMP    OPTION,#4    ;MODIFY OPTION ?
47 027446 001003                BNE   23$           ;BR IF NO
48 027450 005237 002170        INC   MODTD        ;SET MODIFY OPTION
49 027454 000413                BR    26$
50 027456 005237 002174        23$: INC   WRTFE2     ;SET WRITE FE-2 OPTION
51 027462 000403                BR    25$
52
53           ;ENTER HERE TO FORMAT HEADERS & DATA
54
55 027464 012721 000075        24$: MOV    #RDTD,(R1)+ ;LOAD THE READ TD OPERATION

```

```

56 027470 000411      BR      27$
57
58                      ;ENTER HERE TO VERIFY TD(S) & HEADER(S)
59
60 027472 012721 000075 25$:  MOV    #RDTD,(R1)+    ;LOAD THE READ TD OPERATION
61 027476 012721 000073    MOV    #RDHD,(R1)+    ;LOAD THE READ HEADER AND DATA FUNCTION
62 027502 000411      BR      29$
63
64                      ;ENTER HERE TO MODIFY TD(S)
65
66 027504 005037 002416 26$:  CLR    MORETD          ;DEFAULT TO NO MORE TD'S TO MODIFY
67                                ;THE READ TD COMMAND IS IMBEDDED IN THE
68                                ;'INPUTD' SUBROUTINE
69 027510 012721 000065    MOV    #WRTTD,(R1)+    ;LOAD THE WRITE TD OPERATION
70 027514 012721 000063 27$:  MOV    #FMTRK,(R1)+    ;LOAD THE FORMAT TRACK OPERATION
71 027520 000402      BR      29$
72
73                      ;ENTER HERE TO LIST TD(S) WITH DEFECTS
74
75 027522 012721 000075 28$:  MOV    #RDTD,(R1)+    ;LOAD THE READ TD OPERATION
76
77                      ;ENTER HERE TO SCAN FOR NEW DEFECTS
78
79 027526 022701 002612 29$:  CMP    #CMDQUE+10.,R1  ;QUEUE FULL?
80 027532 103402      BLO    30$            ;TAKE BRANCH IF SO
81 027534 005021      CLR    (R1)+          ;ZERO THE BALANCE OF THE QUEUE
82 027536 000773      BR      29$            ;AND DO UNTIL QUEUE IS DONE
83 027540 012601 30$:  MOV    (SP)+,R1          ;:POP STACK INTO R1
84
108
109 027542 104432      TRAP   C$EXIT
110 027544 000002      .WORD  L10011-.
111
112                      .EVEN
113
124 027546 104411  L10011: TRAP   C$INIT
027546

```

1
2
3
4
5
6
7
8
9
10 027550
11
18 027550
027550 104461

.SBTTL AUTODROP SECTION

; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
; DROPPED FROM TESTING.

L\$AUTO::

L10012: TRAP C\$AUTO

```

1          .SBTTL  CLEANUP CODING SECTION
2
3          ;**
4          ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
5          ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
6          ;--
7
8 027552    L$CLEAN::
9
10 027552   012700   000340           MOV     @PRI07,R0           ;SET PRIORITY TO 7
      027556   104441           TRAP    C$SPRI
11 027560   012777   000040   152662   MOV     @CLR,@RPCS2       ;MASSBUS INIT TO CLEAR IMPENDING INTERRUPTS
12 027566   013777   002434   152654   MOV     DRVNO,@RPCS2     ;GET DRIVE NUMBER
13
14 027574   012746   002656           MOV     @CRLF,-(SP)      ;CR-LF
      027600   012746   000001           MOV     @1,-(SP)
      027604   010600           MOV     SP,R0
      027606   104417           TRAP    C$PNTF
      027610   062706   000004           ADD     @4,SP
15 027614   005737   002114           TST     L$TEST           ;DID PROGRAM ENTER TEST 1 ?
16 027620   001002           BNE     1$              ;BR IF YES
17 027622   104432           TRAP    C$EXIT
      027624   000232           .WORD   L10013..
18
19 027626   012701   000001           1$:   MOV     @1,R1           ;R1 = 1
20
21 027632   013746   002302           MOV     DEF1,-(SP)      ;PRINT 'TRACKS WITH 1 DEFECTS='
      027636   010146           MOV     R1,-(SP)
      027640   012746   003532           MOV     @FRMT12,-(SP)
      027644   012746   000003           MOV     @3,-(SP)
      027650   010600           MOV     SP,R0
      027652   104417           TRAP    C$PNTF
      027654   062706   000010           ADD     @10,SP
22 027660   005201           INC     R1              ;R1 = 2
23
24 027662   013746   002304           MOV     DEF2,-(SP)      ;PRINT 'TRACKS WITH 2 DEFECTS='
      027666   010146           MOV     R1,-(SP)
      027670   012746   003532           MOV     @FRMT12,-(SP)
      027674   012746   000003           MOV     @3,-(SP)
      027700   010600           MOV     SP,R0
      027702   104417           TRAP    C$PNTF
      027704   062706   000010           ADD     @10,SP
25 027710   005201           INC     R1              ;R1 = 3
26
27 027712   013746   002306           MOV     DEF3,-(SP)      ;PRINT 'TRACKS WITH 3 DEFECTS='
      027716   010146           MOV     R1,-(SP)
      027720   012746   003532           MOV     @FRMT12,-(SP)
      027724   012746   000003           MOV     @3,-(SP)
      027730   010600           MOV     SP,R0
      027732   104417           TRAP    C$PNTF
      027734   062706   000010           ADD     @10,SP
28 027740   005737   002310           TST     DEF4           ;WERE THERE ANY TRACKS WITH 4 DEFECTS ?
29 027744   001414           BEQ     2$              ;BR IF NO
30
31
32 027746   013746   002310           MOV     DEF4,-(SP)
      027752   012746   006206           MOV     @WARN,-(SP)

```

```

027756 012746 005157      MOV      #FRMT62,-(SP)
027762 012746 000003      MOV      #3,-(SP)
027766 010600              MOV      SP,RO
027770 104417              TRAP    C:PNTF
027772 062706 000010      ADD      #10,SP
33 027776                2$:      ;PRINT 'TOTAL DEFECTS FOUND= X.'
34 027776 013746 002254      MOV      DEFCNT,-(SP)
030002 012746 003600      MOV      #FRMT13,-(SP)
030006 012746 000002      MOV      #2,-(SP)
030012 010600              MOV      SP,RO
030014 104417              TRAP    C:PNTF
030016 062706 000006      ADD      #6,SP
35                          ;PRINT 'DONE,  RETRIES MADE= X.,  ERRORS DETECTED= X'
36 030022 013746 002266      MOV      ERTTL,-(SP)
030026 013746 002270      MOV      ERRITL,-(SP)
030032 012746 003437      MOV      #FRMT11,-(SP)
030036 012746 000003      MOV      #3,-(SP)
030042 010600              MOV      SP,RO
030044 104417              TRAP    C:PNTF
030046 062706 000010      ADD      #10,SP
37
46
47 030052 104432              TRAP    C:EXIT
030054 000002              .WORD  L10013-.
48
60                          .EVEN
61
62 030056                L10013:
030056 104412              TRAP    C:CLEAN

```

```

1          .SBTTL  DROP UNIT SECTION
2
3
4          ;**
5          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
6          ; TO NO LONGER BE TESTED.
7          ;--
8 030060   L$DU::
9
18
19 030060 000167      .WORD  J$JMP
   030062 000000      .WORD  L10014-2-.
20
32          .EVEN
33
34 030064      L10014:
   030064 104453      TRAP   C$DU

```

```

1      .SBTTL  ADD UNIT SECTION
2
3
4      ;**
5      ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
6      ; TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
7      ; TO THE TEST CYCLE.
8      ;--
9      L$AU::
10
11
12
13
14
15
16
17
18
19
20     030066  000167      .WORD  J$JMP
21     030070  000000      .WORD  L10015-2-.
22
23
24
25
26
27
28
29
30
31
32
33
34
35     030072  104452      L10015: TRAP  C$AU

```


118	030316	001005			BNE	9#			;BR IF NO
119	030320	104450			TRAP	C#MANI			
120	030322	103003			BCC	9#			
121	030324	004737	012006		JSR	PC,INPUTD			;CALL THE INPUT TD SUBROUTINE
122									
123	030330	005725		8#:	TST	(R5)+			;POINT TO NEXT COMMAND IN QUEUE
124	030332	005715		9#:	TST	(R5)			;ANY MORE COMMANDS IN QUEUE ?
125	030334	001417			BEQ	10#			;BR IF NO
126	030336	012537	002404		MOV	(R5)+,FUNCTN			;THIS IS A FORMAT OR VERIFY COMMAND
127	030342	004737	015452		JSR	PC,MAKTRK			;NOW EXECUTE THE COMMAND
128	030346	000430			BR	12#			;RETURN HERE IF EXCEEDED RETRY LIMIT
129									;ELSE RETURN HERE, NO ERROR
130	030350	022737	000073	002404	CMP	#RDHD,FUNCTN			;IS THIS A VERIFY COMMAND ?
131	030356	001006			BNE	10#			;BR IF NO
132	030360	004737	017614		JSR	PC,VERIFY			;ANY MIS-COMPARES DURING HEADER VERIFY ?
133	030364	001403			BEQ	10#			;BR IF NO
134	030366	004737	013402		JSR	PC,ERRORS			;REPORT THE ERROR
135	030372	000416			BR	12#			
136									
137	030374	005737	002172	10#:	TST	LIST			;IS LIST OPTION ENABLED ?
138	030400	001402			BEQ	11#			;BR IF NO
139	030402	004737	021714		JSR	PC,LISTDS			;CALL SUBROUTINE TO LIST THE TD(S) WITH DEFECTS
140	030406	004737	020676	11#:	JSR	PC,CNTDEF			;INCLUDE THIS DATA IN THE DEFECT COUNT
141	030412	005737	002170		TST	MODTD			;IS MODIFY TD MODE ENABLED ?
142	030416	001404			BEQ	12#			;BR IF NO
143	030420	005737	002416		TST	MORETD			;DO WE HAVE ANYMORE TD'S TO MODIFY ?
144	030424	001315			BNE	6#			;BR IF YES
145	030426	000406			BR	13#			
146									
147	030430	004737	020646	12#:	JSR	PC,SPIRAL			;GET THE NEXT SEQUENTIAL DISK ADDRESS
148	030434	023737	002312	002212	CMP	DESCYL,MAXCYL			;DONE ALL CYLINDERS YET ?
149	030442	101706			BLOS	6#			;BR IF NO
150	030444	004737	020160	13#:	JSR	PC,WRTBSF			;WRITE THE BAD SECTOR FILE
151	030450	000137	031126		JMP	EXIT1			;JUMP TO EXIT

```
1          .SBTTL TEST 1: SCAN SECTION
2
3 030454 012737 177777 002252 SCAN:  MOV  #-1,SUPRSS ;SUPPRESS THE ERROR MESSAGE OUTPUT
4 030462 105037 002246          CLR  NEWCNT ;RESET THE LOCAL DEFECT COUNTER AND
5 030466 105037 002242          CLR  FIRPAS ;FIRST PASS OF SCAN FLAG
6 030472 012737 032464 002240      MOV  #DEFBUF,LASLOC ;SAVE LOCATION OF LAST DEFECT IN BUFFER
7 030500 012701 032464          MOV  #DEFBUF,R1 ;GET POINTER TO BEGINNING OF DEFECT BUFFER
8 030504 005021          1$: CLR  (R1)+ ;INITIALIZE DEFECT AND DATA BUFFERS
9 030506 020127 120072          CMP  R1,#ENDBUF ;DONE BUFFERS YET ?
10 030512 101774          BLOS 1$ ;BR IF NO
11 030514 012701 032464          MOV  #DEFBUF,R1 ;GET POINTER TO BEGINNING OF DEFECT BUFFER
12
13          2$: ;>>>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<<<<<<<<
14 030520 104422          TRAP C#BRK
15 030522 013737 002312 002316      MOV  DESCYL,SAVCYL ;SAVE THE DESIRED CYLINDER AND
16 030530 013737 002314 002320      MOV  DESTRK,SAVTRK ;DESIRED TRK/SEC ADDRESSES BEFORE SCANNING.
17
18 030536 004737 022546          3$: JSR  PC,TDSCAN ;LOOK FOR A CORRUPTED TD
19 030542 000407          BR   4$ ;RETURN HERE IF ERROR
20 ;ELSE RETURN HERE, NO ERROR
21
22 030544 004737 022732          JSR  PC;HDSCAN ;LOOK FOR A CORRUPTED HEADER
23 030550 000404          BR   4$ ;RETURN HERE IF ERROR
24 ;ELSE RETURN HERE, NO ERROR
25
26 030552 004737 023406          JSR  PC,DASCAN ;LOOK FOR DATA DEFECTS
27 030556 000401          BR   4$ ;RETURN HERE IF ERROR
28 030560 000405          BR   5$ ;ELSE RETURN HERE, NO ERROR
29
30 030562 004737 013402          4$: JSR  PC,ERRORS ;REPORT ERROR, IF SUPRSS=0
31 030566 012737 177777 002252      MOV  #-1,SUPRSS ;SUPPRESS THE ERROR MESSAGE OUTPUT
32
33 030574 105737 002242          5$: TSTB FIRPAS ;IS THIS THE 2ND PASS OF SCAN ?
34 030600 100412          BMI  8$ ;BR IF YES
35 030602 004737 020646          6$: JSR  PC,SPIRAL ;GET THE NEXT SEQUENTIAL DISK ADDRESS
36 030606 105737 002246          7$: TSTB NEWCNT ;IS DEFECT BUFFER FULL ?
37 030612 100435          BMI  10$ ;BR IF YES
38 030614 023737 002312 002212      CMP  DESCYL,MAXCYL ;DONE ALL CYLINDERS YET ?
39 030622 101736          BLOS 2$ ;BR IF NO
40 030624 000430          BR   10$
41
42 030626 004737 020604          8$: JSR  PC,GETNEX ;GET THE NEXT DISK ADDRESS IN DEFECT BUFFER
43 030632 023737 002256 002240      CMP  TBLPTR,LASLOC ;DONE CHECKING THE CORRECTED DEFECTS ?
44 030640 101736          BLOS 3$ ;BR IF NO
45 030642 013701 002240          MOV  LASLOC,R1 ;GET LAST LOCATION IN DEFECT BUFFER AND
46 030646 005021          9$: CLR  (R1)+ ;INITIALIZE REST IT.
47 030650 020127 036072          CMP  R1,#ENDEFF ;DONE BUFFER YET ?
48 030654 101774          BLOS 9$ ;BR IF NO
49 030656 105737 002246          TSTB NEWCNT ;ANY NEW DEFECTS FOUND ?
50 030662 001014          BNE  11$ ;BR IF YES
51 030664 013737 002316 002312      MOV  SAVCYL,DESCYL ;RESTORE THE DESIRED CYLINDER AND
52 030672 013737 002320 002314      MOV  SAVTRK,DESTRK ;TRK/SEC ADDRESSES, THEN
53 030700 004737 020646          JSR  PC,SPIRAL ;GET THE NEXT SEQUENTIAL DISK ADDRESS.
54 030704 000500          BR   19$
55
56 ;SEE IF THERE WERE ANY DEFECTS DURING THE SCAN, IF SO, LOOK FOR ANY
57 ;POSSIBLE SCRATCHES AND FLAG THEM. WRITE THE NEW DEFECT(S) INTO THE
```

TEST 1: SCAN SECTION

```
58                                     ;CORRESPONDING TRACK DESCRIPTOR(S) IF ENABLED.
59
60 030706 105737 002246                 10#: TSTB   NEWCNT      ;ANY NEW DEFECTS FOUND ?
61 030712 001475                        BEQ     19#           ;BR IF ZERO, NO
62 030714 012701 036074                 11#: MOV    #DBUFF,R1  ;GET POINTER TO BEGINNING OF DATA BUFFER AND
63 030720 005021                        12#: CLR    (R1)+     ;INITIALIZE IT.
64 030722 020127 120072                 CMP    R1,#ENDBUF   ;DONE BUFFER YET ?
65 030726 101774                        BLOS   12#          ;BR IF NO
66 030730 004737 025312                 JSR    PC,FINSCR    ;LOOK FOR SCRATCHES IN THIS AREA;
67                                        ;PUT THE CONTENTS OF 'DEFBUF' (INCLUDING ANY
68                                        ;INTERPOLATED SCRATCHES) INTO 'DBUFF'.
69 030734 012701 036074                 MOV    #DBUFF,R1   ;GET POINTER TO BEGINNING OF DATA BUFFER
70 030740 004737 022224                 JSR    PC,NEWLST   ;PRINT NEW DEFECTS, IF ANY
71 030744 005737 002200                 TST   ENWTTD      ;IS WRITE TD(S) ENABLED ?
72 030750 001456                        BEQ    19#         ;BR IF NO
73 030752 013700 002226                 13#: MOV    ERRMAX,RO ;SETUP RETRY COUNT
74 030756 004737 026162                 JSR    PC,LOADTD  ;LOAD THE TD FOR THE CORRECT TRACK AND
75 030762 001426                        BEQ    17#         ;BRANCH IF NO DEFECT STATUS
76 030764 004737 026236                 JSR    PC,SAVETD  ;SAVE TO WRITE BUFFER JUST IN CASE THERE
77                                        ;IS RETRY ATTEMPT.
78 030770 004737 026274                 14#: JSR    PC,RESTTD ;NOW RESTORE THE TD BUFFER
79 030774 004737 021152                 JSR    PC,WRITTD  ;CALL THE WRITE TD SUBROUTINE
80 031000 000417                        BR     17#        ;RETURN HERE IF EXCEEDED RETRY LIMIT
81                                        ;ELSE RETURN HERE, NO ERROR
82 031002 004737 021440                 JSR    PC,READTD  ;CALL THE READ TD SUBROUTINE
83 031006 000401                        BR     15#        ;RETURN HERE IF EXCEEDED RETRY LIMIT
84 031010 000403                        BR     16#        ;ELSE RETURN HERE, NO ERROR
85 031012 005300                 15#: DEC    RO     ;DID WE EXCEED RETRY ?
86 031014 001411                        BEQ    17#        ;BR IF YES
87 031016 000764                        BR     14#        ;TRY WRITTING TD AGAIN
88
89 031020 012737 000063 002404          16#: MOV    #FMTRK,FUNCTN ;LOAD THE FORMAT COMMAND FOR THE DRIVER
90 031026 004737 015452                 JSR    PC,MAKTRK  ;NOW EXECUTE THE COMMAND
91 031032 000402                        BR     17#        ;RETURN HERE IF EXCEEDED RETRY LIMIT
92                                        ;ELSE RETURN HERE, NO ERROR
93 031034 004737 020676                 JSR    PC,CNTDEF  ;COUNT THESE NEW DEFECTS IN THE OVERALL COUNTER
94 031040 005711                 17#: TST   (R1)    ;IS THIS END OF DEFECTS IN BUFFER ?
95 031042 001403                        BEQ    18#        ;BR IF YES
96 031044 020127 120072                 CMP    R1,#ENDBUF ;AT END OF THE SCRATCH BUFFER ?
97 031050 101740                        BLOS   13#        ;BR IF NO
98
99                                     ;NOW GET THE TRACK DESCRIPTORS JUST WRITTEN AND GO SCAN THEM FOR
100                                    ;ANYMORE POSSIBLE DEFECTS AFTER THE CORRECTION.
101
102 031052 105037 002246                 18#: CLRB  NEWCNT      ;RESET THE LOCAL DEFECT COUNTER
103 031056 004737 026404                 JSR    PC,RESTOR  ;NOW RESTORE THE ORIGINAL DEFECT BUFFER;
104                                        ;PUT THE CONTENTS OF 'DBUFF' (EXCLUDING THE
105                                        ;INTERPOLATED SCRATCHES) INTO 'DEFBUF'.
106 031062 012737 032464 002256          MOV    #DEFBUF,TBLPTR ;RESTORE POINTER TO BEGINNING OF DEFECT BUFFER
107 031070 004737 020604                 JSR    PC,GETNEX  ;GET NEXT DEFECT TO CHECK
108 031074 112737 177777 002242          MOVB  #-1,FIRPAS   ;INDICATE 2ND PASS OF SCANNER
109 031102 000137 030536                 JMP    3#         ;DO SOME MORE SCANNING
110
111 031106 023737 002312 002212          19#: CMP    DESCYL,MAXCYL ;DONE ALL CYLINDERS YET ?
112 031114 101002                        BHI   20#         ;BR IF YES
113 031116 000137 030454                 JMP    SCAN       ;DO SOME MORE SCANNING
114
```

```
115 031122 004737 020160      20#:   JSR   PC,WRTBSF      ;WRITE THE BAD SECTOR FILE
116
117 031126                    EXIT1:
123
135                          .EVEN
136
137 031125                    L10016:
    031126 104401            TRAP   C#ETST
138
```

2
13
14
42
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
67
68
69
70
71
72
76
86

```

.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.
; --

        .WORD L10017-L$HARD/2
L$HARD::                                ;PRINT 'RPCS1 ADRS?'
        .WORD  T$CODE
        .WORD  MSG1
        .WORD  T$LOLIM
        .WORD  T$HILIM
                                           ;PRINT 'VECTOR ADRS?'
        .WORD  T$CODE
        .WORD  MSG4
        .WORD  T$LOLIM
        .WORD  T$HILIM
                                           ;PRINT 'BR LEVEL?'
        .WORD  T$CODE
        .WORD  MSG5
        .WORD  340
        .WORD  T$LOLIM
        .WORD  T$HILIM
                                           ;PRINT 'DRIVE #?'
        .WORD  T$CODE
        .WORD  MSG6
        .WORD  7
        .WORD  T$LOLIM
        .WORD  T$HILIM
        .EVEN
L10017:
        .ASCIZ  /RPCS1 ADRS/
        .ASCIZ  /VECTOR ADRS/
        .ASCIZ  /BR LEVEL/
        .ASCIZ  /DRIVE #/
        .EVEN

```

031130	000022			
031132				
031132	000031			
031134	031176			
031136	160000			
031140	177777			
031142	001031			
031144	031211			
031146	000000			
031150	000377			
031152	002032			
031154	031225			
031156	000340			
031160	000000			
031162	000007			
031164	003032			
031166	031236			
031170	000007			
031172	000000			
031174	000007			
031176				
031176	122	120	103	MSG1:
031211	126	105	103	MSG4:
031225	102	122	040	MSG5:
031236	104	122	111	MSG6:

```

1      .SBTTL  SOFTWARE PARAMETER CODING SECTION
2
3
4      ;**
5      ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
6      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
7      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
9      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
10     ; WITH THE OPERATOR.
11     ;--
12     031246 000000      .WORD L10020-L$SOFT/2
13     031250
22     031250      .EVEN
23     L10020:
30
31     031250      $PATCH::.BLKW 50.      ;PROGRAM PATCH AREA (50. WORDS)
32
38     031414 000000      TDWRD1::.WORD 0      ;TD WORD #1
39     031416 000000      TDWRD2::.WORD 0      ;TD WORD #2
40     031420 000000      TDWRD3::.WORD 0      ;TD WORD #3
41     031422 000000      TDWRD4::.WORD 0      ;TD WORD #4
42     031424 000000      TDWRD5::.WORD 0      ;TD WORD #5
43     031426 000000      TDWRD6::.WORD 0      ;TD WORD #6
44
48     031430 000000      TDCPY1::.WORD 0      ;COPY OF TD WORD #1
49     031432 000000      TDCPY2::.WORD 0      ;COPY OF TD WORD #2
50     031434 000000      TDCPY3::.WORD 0      ;COPY OF TD WORD #3
51     031436 000000      TDCPY4::.WORD 0      ;COPY OF TD WORD #4
52     031440 000000      TDCPY5::.WORD 0      ;COPY OF TD WORD #5
53     031442 000000      TDCPY6::.WORD 0      ;COPY OF TD WORD #6
54
58     031444 000000      TDTMP1::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #1
59     031446 000000      TDTMP2::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #2
60     031450 000000      TDTMP3::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #3
61     031452 000000      TDTMP4::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #4
62     031454 000000      TDTMP5::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #5
63     031456 000000      TDTMP6::.WORD 0      ;TEMPORARY STORAGE FOR TD WORD #6
64
62     031460      BEGBUF = .      ;BEGINNING OF BUFFER SECTION
63     031460      TMPBSF::.BLKW 2      ;GENERAL BSF USE
64     031464      MFGBUF::.BLKW 256.      ;MFG BUFFER FOR DEC 144 FILE
65     032462      ENDBSF = .-2      ;END OF BSF BUFFER
66
68     032464      DEFBUF = .      ;BEGINNING OF DEFECT BUFFER (SCAN MODE)
69     032464      HDRBLK::.BLKW 6*50.      ;HEADER BLOCK (6 WRDS X 50. SECTORS)
70     033614      TDSBLK::.BLKW 6*50.      ;TRACK DESCRIPTOR BLOCK
71     034744      LINBUF::.BLKW 6*50.      ;BEGINNING OF LINEAR BUFFER
72     036072      ENDLIN = .-2      ;END OF LINEAR BUFFER
73     036072      ENDDF = ENDLIN      ;END OF DEFECT BUFFER
74
75     036074      DBUFF::.BLKW 256.*50.      ;DATA BUFFER (256. WRDS X 50. SECTORS)
76     120072      ENDBUF = .-2      ;END OF BUFFER SECTION
77
84     .EVEN

```

120074 120114
120076 000006
120100

.WORD T\$FREE
.WORD T\$SIZE
L\$LAST::

1				
14				
16	120100	000000	.WORD	0
	120102	000004	.WORD	L10023-./2-1
	120104			
17	120104	176700	.WORD	176700
18	120106	000254	.WORD	254
19	120110	000240	.WORD	240
20	120112	000000	.WORD	0
21	120114			
23		000001		
			L10023:	
			.END	

FMT16	12-154#	25-15														
FMTRK	12-220#	17-92	33-25	72-70	78-89											
FORMAT	14-98#	33-27														
FORMT	13-23#	72-18	72-32*													
FRMT00	14-41#	19-16														
FRMT01	14-42#	19-17														
FRMT02	14-43#	19-19														
FRMT03	14-44#	19-20														
FRMT04	14-45#	19-22														
FRMT05	14-46#	19-23														
FRMT06	14-47#	19-28														
FRMT07	14-48#	19-29														
FRMT10	14-50#	34-13														
FRMT11	14-51#	74-36														
FRMT12	14-52#	74-21	74-24	74-27												
FRMT13	14-53#	74-34														
FRMT14	14-54#	39-39	51-48	52-45												
FRMT15	14-55#	39-42	51-51	52-48												
FRMT16	14-56#	15-30														
FRMT17	14-57#	15-31														
FRMT20	14-59#	18-20														
FRMT22	14-60#	18-16														
FRMT23	14-61#	15-19	15-21													
FRMT24	14-62#	23-50														
FRMT30	14-64#	34-28														
FRMT31	14-65#	34-25														
FRMT32	14-66#	52-21														
FRMT33	14-67#	15-45														
FRMT35	14-68#	39-22	52-24													
FRMT36	14-69#	39-33	52-41													
FRMT37	14-70#	51-43														
FRMT50	14-72#	71-37														
FRMT51	14-76#	71-39														
FRMT60	14-82#	38-57														
FRMT61	14-83#	38-45	47-38													
FRMT62	14-84#	74-32														
FRMT63	14-85#	20-38														
FRMT64	14-86#	42-38														
FRMT65	14-87#	38-61														
FRMT66	14-88#	38-63														
FRMTT	14-40#	39-20														
FUNCTN	13-128#	17-47	17-62	17-73	17-92	22-19*	33-25	33-30	34-83*	42-18	42-23*	42-53*	43-23*	49-36*		
	50-19	50-20*	50-63*	51-15	51-26*	51-58*	54-19*	55-19*	55-36*	77-126*	77-130	78-89*				
G#CNT0	7-144#															
G#DELM	7-144#	24-11														
G#DISP	7-144#															
G#EXCP	7-144#															
G#HILI	7-144#															
G#LOLI	7-144#															
G#NO	7-144#	23-71	23-73	23-81	71-61											
G#OFFS	7-144#	23-17	23-23	23-25	23-52	23-56	23-60	23-64	23-71	23-73	23-81	23-118	71-41	71-54		
		71-61	71-70	71-78	71-80	71-85	71-87	71-92	71-98	71-103	80-55	80-57	80-59	80-61		
G#OFSI	7-144#	23-17	23-23	23-25	23-52	23-56	23-60	23-64	23-71	23-73	23-81	23-118	71-41	71-54		
		71-61	71-70	71-78	71-80	71-85	71-87	71-92	71-98	71-103	80-55	80-57	80-59	80-61		
G#PRMA	7-144#	80-55	80-57													
G#PRMD	7-144#	23-17	23-23	23-25	23-52	23-56	23-60	23-64	23-71	23-73	23-81	71-41	71-78	71-80		

ILEV	12-63#	34-21						
ILF	12-78#							
ILR	12-79#							
INPUTD	23-15#	77-121						
INSERT	23-82	38-16#	54-82	55-78	64-20			
INTEN	12-5#							
INTLEV	13-55#	22-15	32-36	34-23*	36-15	41-18	54-61	
IR	12-31#							
ISR	11-57#							
ITCNT	13-66#	26-21*	53-14*	53-21	54-15*	54-36	55-33*	55-51
IXE	11-57#							
IXU	12-174#							
J\$JMP	7-144#	69-61	75-19	76-20				
L\$ACP	7-198#							
L\$APT	7-198#							
L\$AU	76-9#							
L\$AUT	7-198#							
L\$AUTO	7-198	73-10#						
L\$CCP	7-198#							
L\$CLEA	7-198	74-8#						
L\$CO	7-198#							
L\$DEPO	7-198#							
L\$DESC	7-198	14-27#						
L\$DESP	7-198#							
L\$DEVP	7-198#							
L\$DISP	7-198	8-8#						
L\$DLY	7-198#	24-11						
L\$DTP	7-198#							
L\$DTYP	7-198#							
L\$DU	75-8#							
L\$DUT	7-198#							
L\$DVTY	7-198	14-17#						
L\$EF	7-198#							
L\$ENVI	7-198#							
L\$ETP	7-198#							
L\$EXP1	7-198#							
L\$EXP4	7-198#							
L\$EXPS	7-198#							
L\$HARD	7-198	80-53	80-53#					
L\$HIME	7-198#							
L\$HPCP	7-198#							
L\$HPTP	7-198#							
L\$HM	7-198	9-10	9-10#					
L\$ICP	7-198#							
L\$INIT	7-198	71-8#						
L\$LADP	7-198#							
L\$LAST	7-198	81-84#	82-22					
L\$LOAD	7-198#							
L\$LUN	7-198#							
L\$MREV	7-198#							
L\$NAME	7-198#							
L\$PRIO	7-198#							
L\$PROT	7-198	70-8#						
L\$PRT	7-198#							
L\$REPP	7-198#							
L\$REV	7-198#							

0%GNSW	7-144#	7-198												
0%POIN	7-144#	7-176	7-176#	7-198										
0%SETU	7-144#	7-176#	7-198	81-84										
OFFDIR	12-151#													
OFFSET	13-50#	17-40	22-21*	25-15*	33-38*	49-39*	49-54*	50-29*	54-23*					
OM	12-61#													
ONEFIL	2-4#	2-8	4-58	5-1	7-129#	7-165	10-22	11-1	11-8#	11-13	68-2	69-1	69-8#	69-13
OPI	76-37	77-1	77-9#	77-14	79-2	80-1	80-9#	80-15						
OPTION	12-91#	54-30	54-63											
	13-21#	71-41	71-45	71-47	71-49	71-63	71-65	71-89	71-94	71-100	72-30	72-34	72-38	72-42
	72-46													
OR	12-32#													
PAR	12-81#													
PARTBL	13-97#	30-72												
PAT	12-29#													
PATRN	13-86#	55-15*	55-53	58-14	58-23*									
PGE	12-35#	12-193#												
PGM	12-67#													
PHF	12-176#													
PIP	12-71#													
PLTRK	13-84#	32-16	36-18	54-59										
PNT	11-57#													
PRI	11-57#													
PRI00	11-57#													
PRI01	11-57#													
PRI02	11-57#													
PRI03	11-57#													
PRI04	11-57#													
PRI05	11-57#													
PRI06	11-57#													
PRI07	11-57#	72-13	74-10											
PSEL	12-9#													
RCVED	14-104#	15-21												
RDBSF	42-17#	77-95												
RDDAT	12-222#	42-23												
RDND	12-223#	17-62	33-30	51-26	54-19	72-61	77-130							
RDTD	12-224#	50-20	72-55	72-60	72-75	77-111								
RDY	12-6#	17-49	17-50											
READIN	12-214#	25-14												
READTD	23-27	23-108	50-19#	53-16	64-15	77-113	78-82							
READY	17-26#	17-27	17-35	25-16										
RECAL	12-207#	34-73	34-83											
RECALI	14-97#	34-72												
RECVED	13-72#	15-45	41-36*											
REG	13-168#	19-17	19-17	19-17	19-17	19-17	19-17	19-20	19-20	19-20	19-20	19-20	19-20	19-20
	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-29	19-29	40-16	54-30	54-63	55-45	56-18
	56-24	56-48												
RELSE	12-209#													
RELTV	23-74	37-14#	54-74	55-66										
RELWRD	13-126#	23-73	23-75	23-77*	23-78*	23-81	37-16	37-25*	38-30*	38-31	38-33*	38-34*	38-61	38-66*
	38-77	54-50*	54-55*	54-71*	54-75	55-67	55-74	55-76*	55-77*	56-30*	57-36*	60-16*	60-22	60-25*
	64-19	64-21*	65-100*											
RESET	25-13#	26-37	34-11	42-52										
RESTOR	67-7#	78-103												
RESTD	23-99	66-65#	78-78											
RETRY	26-21#	33-40	34-52	34-66	34-77	34-85	42-26	43-27	43-43	49-42	50-34	54-25	55-24	55-40

71-37	71-37	71-37	71-37	71-39	71-39	71-39	71-39	71-39	71-39	71-39	71-39	71-39	71-39
71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41	71-41
71-41	71-41	71-54	71-54	71-54	71-54	71-54	71-54	71-54	71-54	71-54	71-54	71-54	71-54
71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-61	71-70	71-70
71-70	71-70	71-70	71-70	71-70	71-70	71-70	71-70	71-70	71-70	71-70	71-78	71-78	71-78
71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-78	71-80
71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-80
71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85	71-85
71-85	71-85	71-87	71-87	71-87	71-87	71-87	71-87	71-87	71-87	71-87	71-87	71-87	71-87
71-87	71-87	71-87	71-87	71-92	71-92	71-92	71-92	71-92	71-92	71-92	71-92	71-92	71-92
71-92	71-92	71-98	71-98	71-98	71-98	71-98	71-98	71-98	71-98	71-98	71-98	71-98	71-98
71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	71-103	72-10
72-11	72-11	72-11	72-11	72-13	72-13	72-13	72-13	72-13	72-109	72-109	72-109	72-109	72-124
73-18	73-18	74-10	74-10	74-10	74-10	74-14	74-14	74-14	74-14	74-14	74-14	74-14	74-14
74-14	74-14	74-17	74-17	74-17	74-17	74-21	74-21	74-21	74-21	74-21	74-21	74-21	74-21
74-21	74-21	74-21	74-21	74-21	74-21	74-24	74-24	74-24	74-24	74-24	74-24	74-24	74-24
74-24	74-24	74-24	74-24	74-24	74-24	74-27	74-27	74-27	74-27	74-27	74-27	74-27	74-27
74-27	74-27	74-27	74-27	74-27	74-27	74-32	74-32	74-32	74-32	74-32	74-32	74-32	74-32
74-32	74-32	74-32	74-32	74-32	74-32	74-34	74-34	74-34	74-34	74-34	74-34	74-34	74-34
74-34	74-34	74-34	74-34	74-36	74-36	74-36	74-36	74-36	74-36	74-36	74-36	74-36	74-36
74-36	74-36	74-36	74-36	74-47	74-47	74-47	74-47	74-47	74-62	74-62	75-19	75-19	75-19
75-34	75-34	76-20	76-20	76-20	76-20	76-35	76-35	77-106	77-106	77-119	77-119	77-120	77-120
78-14	78-14	78-137	78-137	80-53	80-53	80-55	80-55	80-55	80-55	80-55	80-55	80-55	80-55
80-57	80-57	80-57	80-57	80-57	80-57	80-57	80-57	80-59	80-59	80-59	80-59	80-59	80-59
80-59	80-59	80-59	80-59	80-61	80-61	80-61	80-61	80-61	80-61	80-61	80-61	80-61	80-61
80-62	80-62	81-12	81-12	81-22	81-22	81-84	81-84	81-84	81-84	81-84	81-84	81-84	82-16
82-16	82-16												
SVCSUB	7-144#	7-152#											
SVCTAG	7-144#	7-154#	9-25	9-25	9-25	10-20	10-20	10-20	15-14	15-14	15-14	15-25	15-25
	15-34	15-34	15-34	15-40	15-40	15-40	15-49	15-49	15-49	23-17	23-17	23-17	23-23
	23-23	23-25	23-25	23-25	23-52	23-52	23-52	23-56	23-56	23-56	23-60	23-60	23-64
	23-64	23-64	23-71	23-71	23-71	23-73	23-73	23-73	23-81	23-81	23-81	23-118	23-118
	69-76	69-76	69-76	71-41	71-41	71-41	71-54	71-54	71-54	71-61	71-61	71-70	71-70
	71-70	71-78	71-78	71-78	71-80	71-80	71-80	71-85	71-85	71-87	71-87	71-87	71-92
	71-92	71-92	71-98	71-98	71-98	71-103	71-103	71-103	72-124	72-124	72-124	73-18	73-18
	74-62	74-62	74-62	75-34	75-34	75-34	76-35	76-35	76-35	78-137	78-137	80-62	80-62
	80-62	81-22	81-22	81-22	82-16	82-16	82-16	82-21	82-21	82-21	82-21		
SVCTST	7-144#	7-151#	77-66	77-66	77-66								
T\$#AU	76-9#	76-20	76-35										
T\$#AUT	73-10#	73-18											
T\$#CLE	74-8#	74-17	74-47	74-62									
T\$#DAT	82-16	82-16#	82-21										
T\$#DU	75-8#	75-19	75-34										
T\$#HAR	80-53	80-53#	80-62										
T\$#HW	9-10	9-10#	9-25										
T\$#INI	71-8#	71-16	71-22	72-11	72-109	72-124							
T\$#MSG	15-9#	15-14	15-16#	15-25	15-27#	15-34	15-36#	15-40	15-42#	15-49			
T\$#PC	82-15#	82-22											
T\$#PRO	70-8#												
T\$#PTA	82-15#	82-16	82-16#										
T\$#RPT	69-47#	69-61	69-76										
T\$#SOF	81-12	81-12#	81-22										
T\$#SW	10-10	10-10#	10-20										
T\$#TES	77-66#	78-137											
T\$#ARGC	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198#	7-198#	7-198#
	7-198#	7-198#	7-198#	15-13	15-13	15-13#	15-19	15-19	15-19	15-19	15-19#	15-19#	15-19#

T\$LTNO	76-35	78-137	80-62	81-22										
T\$NEST	81-84	7-170	7-170	7-170	9-10	9-10	9-10	9-25	9-25	9-25	9-25	10-10	10-10	10-10
	10-20	10-20	10-20	10-20	10-21	10-21	10-21	10-21	11-51	11-51	11-51	15-9	15-9	15-9
	15-14	15-14	15-14	15-14	15-16	15-16	15-16	15-25	15-25	15-25	15-25	15-27	15-27	15-27
	15-34	15-34	15-34	15-34	15-36	15-36	15-36	15-40	15-40	15-40	15-40	15-42	15-42	15-42
	15-49	15-49	15-49	15-49	68-1	68-1	68-1	68-1	69-41	69-41	69-41	69-47	69-47	69-47
	69-76	69-76	69-76	69-76	70-8	70-8	70-8	70-12	70-12	70-12	70-12	71-8	71-8	71-8
	72-124	72-124	72-124	72-124	73-10	73-10	73-10	73-18	73-18	73-18	73-18	74-8	74-8	74-8
	74-62	74-62	74-62	74-62	75-8	75-8	75-8	75-34	75-34	75-34	75-34	76-9	76-9	76-9
	76-35	76-35	76-35	76-35	76-36	76-36	76-36	76-36	77-50	77-50	77-50	77-66	77-66	77-66
	78-137	78-137	78-137	78-137	79-1	79-1	79-1	79-1	80-43	80-43	80-43	80-53	80-53	80-53
	80-62	80-62	80-62	80-62	81-12	81-12	81-12	81-22	81-22	81-22	81-22	81-85	81-85	81-85
	81-85													
T\$NSO	7-170	10-21	11-51	68-1	69-41	76-36	77-50	79-1	80-43	81-85				
T\$NSI	9-10	9-25	10-10	10-20	15-9	15-14	15-16	15-25	15-27	15-34	15-36	15-40	15-42	15-49
	69-47	69-76	70-8	70-12	71-8	72-124	73-10	73-18	74-8	74-62	75-8	75-34	76-9	76-35
	77-66	78-137	80-53	80-62	81-12	81-22								
T\$PCNT	82-15	82-16	82-16	82-16										
T\$PTAB	82-16	82-16												
T\$PTHV	7-198	82-22												
T\$PTNU	7-144	82-16	82-16	82-22	82-22									
T\$SAVL	7-144													
T\$SEGL	7-144													
T\$SIZE	81-84	82-22												
T\$SUBN	7-144	77-66												
T\$TAGL	7-144													
T\$TAGN	7-144	9-10	9-10	9-10	10-10	10-10	10-10	15-9	15-9	15-9	15-16	15-16	15-16	15-27
	15-27	15-27	15-36	15-36	15-42	15-42	15-42	69-47	69-47	69-47	70-8	70-8	70-8	70-8
	71-8	71-8	71-8	73-10	73-10	73-10	74-8	74-8	74-8	75-8	75-8	76-9	76-9	76-9
	76-9	77-66	77-66	77-66	80-53	80-53	81-12	81-12	81-12	81-12	82-15	82-15	82-15	82-16
	82-16	82-16	82-16	82-16	82-16	82-16								
T\$TEMP	8-8	8-8	8-8	8-8	9-25	9-25	10-20	10-20	10-21	10-21	15-14	15-14	15-25	15-25
	15-34	15-34	15-40	15-40	15-49	15-49	23-17	23-17	23-17	23-17	23-17	23-17	23-23	23-23
	23-23	23-23	23-23	23-23	23-25	23-25	23-25	23-25	23-25	23-25	23-52	23-52	23-52	23-52
	23-52	23-52	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-60	23-60	23-60	23-60	23-60
	23-64	23-64	23-64	23-64	23-64	23-64	23-71	23-71	23-71	23-71	23-71	23-71	23-73	23-73
	23-73	23-73	23-73	23-73	23-81	23-81	23-81	23-81	23-81	23-81	23-118	23-118	23-118	23-118
	23-118	23-118	68-1	68-1	69-61	69-61	69-76	69-76	70-12	70-12	71-16	71-16	71-22	71-22
	71-41	71-41	71-41	71-41	71-41	71-41	71-54	71-54	71-54	71-54	71-54	71-54	71-61	71-61
	71-61	71-61	71-61	71-61	71-70	71-70	71-70	71-70	71-70	71-70	71-78	71-78	71-78	71-78
	71-78	71-78	71-80	71-80	71-80	71-80	71-80	71-80	71-80	71-85	71-85	71-85	71-85	71-85
	71-87	71-87	71-87	71-87	71-87	71-87	71-92	71-92	71-92	71-92	71-92	71-92	71-98	71-98
	71-98	71-98	71-98	71-98	71-103	71-103	71-103	71-103	71-103	71-103	72-11	72-11	72-109	72-109
	72-124	72-124	73-18	73-18	74-17	74-17	74-47	74-47	74-62	74-62	75-19	75-19	75-34	75-34
	76-20	76-20	76-35	76-35	76-36	76-36	78-137	78-137	79-1	79-1	80-55	80-55	80-55	80-55
	80-55	80-55	80-57	80-57	80-57	80-57	80-57	80-57	80-59	80-59	80-59	80-59	80-59	80-59
	80-61	80-61	80-61	80-61	80-61	80-61	80-62	80-62	81-22	81-22	81-85	81-85		
T\$TEST	7-144	77-66	77-66	77-66	81-84									
T\$TSTM	7-144	15-13	15-14	15-19	15-21	15-24	15-25	15-30	15-31	15-33	15-34	15-39	15-40	15-45
	15-48	15-49	18-16	18-20	19-16	19-17	19-19	19-20	19-22	19-23	19-28	19-29	20-38	23-17
	23-23	23-25	23-50	23-52	23-56	23-60	23-64	23-71	23-73	23-81	23-118	27-38	27-42	27-45
	27-48	27-51	27-54	27-57	27-60	27-63	27-66	27-69	27-72	27-75	27-78	27-81	27-84	34-13
	34-18	34-25	34-28	34-34	35-21	38-45	38-57	38-61	38-63	39-20	39-22	39-33	39-39	39-42
	39-46	42-38	47-38	51-43	51-48	51-51	51-55	52-21	52-24	52-41	52-45	52-48	52-53	69-76
	71-10	71-12	71-16	71-18	71-22	71-24	71-28	71-37	71-39	71-41	71-54	71-61	71-70	71-78

UPE	12-38#					
US1	12-25#					
US2	12-26#					
US4	12-27#					
VERIFY	41-17#	77-132				
VRIFY	13-24#	34-58	72-36*			
VV	12-64#					
WAIT	24-11#	24-34	34-40			
WARN	14-109#	20-38	38-45	42-38	47-38	74-32
WCE	12-39#	17-66	55-45			
WCF	12-83#					
WCKD	12-217#	55-36				
WCKDAT	14-95#	55-37				
WCKHD	12-218#	22-19				
WCKHDR	14-96#	22-20				
WLE	12-89#					
WLOCK	14-101#	34-64				
WOR	12-187#					
WRDAT	14-94#	43-24	55-20			
WRITCK	17-94	22-15#				
WRITMG	14-118#	71-54				
WRITTD	23-100	48-19	49-22#	78-79		
WRL	12-69#	34-62				
WRSAFM	14-119#	71-61				
WRTBSF	43-15#	77-150	78-115			
WRTDAT	12-219#	43-23	55-19			
WRTFE2	13-28#	72-20	72-50*	77-90		
WRTTD	12-221#	49-36	72-69	77-117		
WRU	12-186#					
X\$ALWA	7-144#					
X\$FALS	7-144#					
X\$OFFS	7-144#					
X\$TRUE	7-144#					

PARAMETER CODING MACRO V04.00 1-DEC-83 09:58:43 PAGE M-1
 CROSS REFERENCE TABLE (CREF V04.00)

SEQ 0167

BCOMPL	1-15#	7-144#	71-26	71-30				
BERROR	1-19#	7-144#						
BGNAU	1-23#	7-144#	76-9					
BGNAUT	1-31#	7-144#	73-10					
BGNCLN	1-39#	7-144#	74-8					
BGN DU	1-47#	7-144#	75-8					
BGNHRD	1-55#	7-144#	80-53					
BGNHW	1-66#	7-144#	9-10					
BGINI	1-77#	7-144#	71-8					
BGNMOD	1-85#	7-144#	7-170	11-51	69-41	77-50	80-43	
BGNMSG	1-98#	7-144#	15-9	15-16	15-27	15-36	15-42	
BGNPRO	1-106#	7-144#	70-8					
BGNPTA	1-114#	7-144#	82-16					
BGNRPT	1-144#	7-144#	69-47					
BGNSEG	1-152#	7-144#						
BGNSET	1-161#	7-144#	82-15					
BGNSFT	1-182#	7-144#	81-12					
BGNSRV	1-193#	7-144#						
BGNSUB	1-201#	7-144#						
BGNSW	1-225#	7-144#	10-10					
BGNTST	1-236#	7-144#	77-66					
BNCOMP	1-266#	7-144#	35-22	71-14	71-20	77-120		
BNERRO	1-270#	7-144#						
BREAK	1-274#	7-144#	77-106	78-14				
BRESET	1-278#	7-144#	71-10					
CKLOOP	1-282#	7-144#						
CLOCK	1-286#	7-144#						
CLOSE	1-292#	7-144#						
CLRVEC	1-296#	7-144#						
COMMEN	1-301#	7-144#						
DELAY	1-322#	7-144#	24-11					
DESCRI	1-317#	7-144#	14-27					
DEVTyp	1-341#	7-144#	14-17					
DISPAT	1-346#	7-144#	8-8					
DISPLA	1-360#	7-144#						
DOCLN	1-376#	7-144#	72-10					
DODU	1-380#	7-144#						
DORPT	1-385#	7-144#						
ENDAU	1-389#	7-144#	76-35					
ENDAUT	1-401#	7-144#	73-18					
ENDCLN	1-413#	7-144#	74-62					
ENDCOM	1-425#	7-144#						
ENDDU	1-441#	7-144#	75-34					
ENDHRD	1-453#	7-144#	80-62					
ENDHW	1-465#	7-144#	9-25					
ENDINI	1-475#	7-144#	72-124					
ENDMOD	1-487#	7-144#	10-21	68-1	76-36	79-1	81-85	
ENDMSG	1-500#	7-144#	15-14	15-25	15-34	15-40	15-49	
ENDPRO	1-512#	7-144#	70-12					
ENDPTA	1-520#	7-144#	82-21					
ENDRPT	1-529#	7-144#	69-76					
ENDSEG	1-541#	7-144#						
ENDSET	1-555#	7-144#	82-22					
ENDSFT	1-568#	7-144#	81-22					
ENDSRV	1-560#	7-144#						
ENDSUB	1-596#	7-144#						

M\$DECR	1-D29#	7-144#	9-25	9-25#	10-20	10-20#	10-21	10-21#	15-14	15-14#	15-25	15-25#	15-34	15-34#
	15-40	15-40#	15-49	15-49#	68-1	68-1#	69-76	69-76#	70-12	70-12#	72-124	72-124#	73-18	73-18#
	74-62	74-62#	75-34	75-34#	76-35	76-35#	76-36	76-36#	78-137	78-137#	79-1	79-1#	80-62	80-62#
	81-22	81-22#	81-85	81-85#	82-16	82-16#								
M\$DEFA	1-E70#	7-144#	23-17	23-17#	23-23	23-23#	23-25	23-25#	23-52	23-52#	23-56	23-56#	23-60	23-60#
	23-64	23-64#	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	71-41	71-41#	71-54	71-54#
	71-61	71-61#	71-70	71-70#	71-78	71-78#	71-80	71-80#	71-85	71-85#	71-87	71-87#	71-92	71-92#
	71-98	71-98#	71-103	71-103#	80-55	80-55#	80-57	80-57#	80-59	80-59#	80-61	80-61#	80-61#	80-61#
M\$ENDE	1-D74#	7-144#	9-25#	10-20#	10-21#	15-14#	15-25#	15-34#	15-40#	15-49#	68-1#	69-76#	72-124#	73-18#
	74-62#	75-34#	76-35#	76-36#	78-137#	79-1#	80-62#	81-22#	81-85#					
M\$ERRI	1-849#	7-144#	27-38	27-38#	27-42	27-42#	27-45	27-45#	27-48	27-48#	27-51	27-51#	27-54	27-54#
	27-57	27-57#	27-60	27-60#	27-63	27-63#	27-66	27-66#	27-69	27-69#	27-72	27-72#	27-75	27-75#
	27-78	27-78#	27-81	27-81#	27-84	27-84#	34-18	34-18#	34-34	34-34#				
M\$ESCA	1-D06#	7-144#												
M\$ESCS	1-D10#	7-144#												
M\$EXCP	1-E01#	7-144#	23-17	23-17#	23-17#	23-23	23-23	23-23#	23-25	23-25#	23-52	23-52#	23-52	23-52#
	23-56	23-56#	23-60	23-60#	23-60	23-60#	23-64	23-64#	23-71	23-71#	23-71#	23-71#	23-73	23-73#
	23-73#	23-81	23-81	23-81#	71-41	71-41#	71-41#	71-78	71-78#	71-80	71-80#	71-80#	71-80#	71-85
	71-85	71-85#	71-87	71-87#	71-87#	71-87#	80-55	80-55#	80-57	80-57#	80-57#	80-59	80-59#	80-59#
	80-61	80-61#	80-61#											
M\$EXIT	1-D14#	7-144#	69-61#	71-16	71-16#	71-22	71-22#	72-11	72-11#	72-109	72-109#	74-17	74-17#	74-47
	74-47#	75-19#	76-20#											
M\$EXSE	1-D22#	7-144#	69-61#	71-16#	71-22#	72-11#	72-109#	74-17#	74-47#	75-19#	76-20#			
M\$EXTJ	1-D18#	7-144#	69-61	69-61#	71-16#	71-22#	72-11#	72-109#	74-17#	74-47#	75-19	75-19#	76-20	76-20#
M\$GEN	1-D38#	7-144#	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	9-25	9-25#	10-10	10-10	10-10#	10-10#	10-20	10-20#	14-17	14-17#	14-27	14-27#	15-9	15-9#
	15-14	15-14#	15-16	15-16#	15-25	15-25#	15-27	15-27#	15-34	15-34#	15-36	15-36#	15-40	15-40#
	15-42	15-42#	15-49	15-49#	23-17	23-17#	23-23	23-23#	23-25	23-25#	23-52	23-52#	23-56	23-56#
	23-60	23-60#	23-64	23-64#	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	69-47	69-47#
	69-76	69-76#	70-8	70-8#	71-8	71-8#	71-41	71-41#	71-54	71-54#	71-61	71-61#	71-70	71-70#
	71-78	71-78#	71-80	71-80#	71-85	71-85#	71-87	71-87#	71-92	71-92#	71-98	71-98#	71-103	71-103#
	72-124	72-124#	73-10	73-10#	73-18	73-18#	74-8	74-8#	74-62	74-62#	75-8	75-8#	75-34	75-34#
	76-9	76-9#	76-35	76-35#	77-66	77-66#	78-137	78-137#	80-53	80-53#	80-62	80-62#	81-12	81-12#
	81-22	81-22#	81-84	81-84#	82-16	82-16#	82-21	82-21#						
M\$GENB	1-C38#	7-144#	23-17	23-17#	23-23	23-23#	23-25	23-25#	23-52	23-52#	23-56	23-56#	23-60	23-60#
	23-64	23-64#	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	71-41	71-41#	71-54	71-54#
	71-61	71-61#	71-70	71-70#	71-78	71-78#	71-80	71-80#	71-85	71-85#	71-87	71-87#	71-92	71-92#
	71-98	71-98#	71-103	71-103#										
M\$GETS	1-D35#	7-144#	9-25	9-25#	10-20	10-20#	10-21	10-21#	15-14	15-14#	15-25	15-25#	15-34	15-34#
	15-40	15-40#	15-49	15-49#	68-1	68-1#	69-76	69-76#	70-12	70-12#	72-124	72-124#	73-18	73-18#
	74-62	74-62#	75-34	75-34#	76-35	76-35#	76-36	76-36#	78-137	78-137#	79-1	79-1#	80-62	80-62#
	81-22	81-22#	81-85	81-85#										
M\$GETT	1-B77#	7-144#	69-61#	71-16#	71-22#	72-11#	72-109#	74-17#	74-47#	75-19#	76-20#			
M\$GNGB	1-C02#	7-144#	7-170#	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#	7-198#
	10-10	10-10	10-10#	11-51#	14-17	14-17#	14-27	14-27#	15-9	15-9#	15-16	15-16#	15-27	15-27#
	15-36	15-36#	15-42	15-42#	69-41#	69-47	69-47#	70-8	70-8#	71-8	71-8#	73-10	73-10#	74-8

15-42	15-42	15-42	15-42	15-45	15-48	15-49	18-16	18-20	19-16	19-17	19-19	19-20	19-22	
19-23	19-28	19-29	20-38	23-17	23-17	23-17	23-23	23-23	23-23	23-25	23-25	23-25	23-50	
23-52	23-52	23-52	23-56	23-56	23-56	23-60	23-60	23-60	23-64	23-64	23-64	23-71	23-71	
23-71	23-73	23-73	23-73	23-81	23-81	23-81	23-118	23-118	23-118	27-38	27-42	27-45	27-48	
27-51	27-54	27-57	27-60	27-63	27-66	27-69	27-72	27-75	27-78	27-81	27-84	34-13	34-18	
34-25	34-28	34-34	35-21	38-45	38-57	38-61	38-63	39-20	39-22	39-33	39-39	39-42	39-46	
42-38	47-38	51-43	51-48	51-51	51-55	52-21	52-24	52-41	52-45	52-48	52-53	69-41	69-41	
69-47	69-47	69-47	69-47	69-76	70-8	70-8	70-8	70-8	71-8	71-8	71-8	71-8	71-10	
71-12	71-16	71-18	71-22	71-24	71-28	71-37	71-39	71-41	71-41	71-41	71-54	71-54	71-54	
71-61	71-61	71-61	71-70	71-70	71-70	71-78	71-78	71-78	71-80	71-80	71-80	71-85	71-85	
71-85	71-87	71-87	71-87	71-92	71-92	71-92	71-98	71-98	71-98	71-103	71-103	71-103	72-10	
72-11	72-13	72-109	72-124	73-10	73-10	73-10	73-10	73-18	74-8	74-8	74-8	74-8	74-10	
74-14	74-17	74-21	74-24	74-27	74-32	74-34	74-36	74-47	74-62	75-8	75-8	75-8	75-8	
75-34	76-9	76-9	76-9	76-9	76-35	77-50	77-50	77-66	77-66	77-66	77-66	77-66	77-66	
77-106	77-119	78-14	78-137	80-43	80-43	80-53	80-53	80-53	80-53	81-12	81-12	81-12	81-12	
82-15	82-15	82-16	82-16	82-16	82-16									
M\$IOSE	1-A00	7-144												
M\$LDRO	1-C42	7-144	35-21	35-21	71-12	71-12	71-18	71-18	71-24	71-24	71-28	71-28	72-13	72-13
	74-10	74-10												
M\$MASK	1-B71	7-144												
M\$MCHI	1-4	7-144	7-144	7-144										
M\$MCLO	1-B24	7-144	7-144	7-144										
M\$MSK1	1-B77	7-144												
M\$POP	1-B81	7-144	9-25	9-25	10-20	10-20	10-21	10-21	15-14	15-14	15-25	15-25	15-34	15-34
	15-40	15-40	15-49	15-49	68-1	68-1	69-76	69-76	70-12	70-12	72-124	72-124	73-18	73-18
	74-62	74-62	75-34	75-34	76-35	76-35	76-36	76-36	78-137	78-137	79-1	79-1	80-62	80-62
M\$PRIN	1-B36	7-144	81-85	81-85										
	15-33	15-33	15-13	15-13	15-19	15-19	15-21	15-21	15-24	15-24	15-30	15-30	15-31	15-31
	19-17	19-17	15-39	15-39	15-45	15-45	15-48	15-48	18-16	18-16	18-20	18-20	19-16	19-16
	20-38	20-38	19-19	19-19	19-20	19-20	19-22	19-22	19-23	19-23	19-28	19-28	19-29	19-29
	38-61	38-61	23-50	23-50	34-13	34-13	34-25	34-25	34-28	34-28	38-45	38-45	38-57	38-57
	39-46	39-46	38-63	38-63	39-20	39-20	39-22	39-22	39-33	39-33	39-39	39-39	39-42	39-42
	52-21	52-21	42-38	42-38	47-38	47-38	51-43	51-43	51-48	51-48	51-51	51-51	51-55	51-55
	71-39	71-39	52-24	52-24	52-41	52-41	52-45	52-45	52-48	52-48	52-53	52-53	71-37	71-37
	74-36	74-36	52-24	52-24	74-14	74-14	74-21	74-21	74-24	74-24	74-32	74-32	74-34	74-34
M\$PUSH	1-B31	7-144	7-170	7-170	9-10	9-10	10-10	10-10	11-51	11-51	15-9	15-9	15-16	15-16
	15-27	15-27	15-36	15-36	15-42	15-42	69-41	69-41	69-47	69-47	70-8	70-8	71-8	71-8
	73-10	73-10	74-8	74-8	75-8	75-8	76-9	76-9	77-50	77-50	77-66	77-66	80-43	80-43
	80-53	80-53	81-12	81-12										
M\$PUT	1-C72	7-144	15-13	15-13	15-13	15-19	15-19	15-19	15-19	15-19	15-19	15-21	15-21	15-21
	15-21	15-21	15-21	15-24	15-24	15-24	15-30	15-30	15-30	15-31	15-31	15-31	15-31	15-31
	15-31	15-31	15-31	15-31	15-31	15-33	15-33	15-33	15-39	15-39	15-39	15-45	15-45	15-45
	15-45	15-45	15-48	15-48	15-48	18-16	18-16	18-16	18-16	18-20	18-20	18-20	18-20	18-20
	19-16	19-16	19-16	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-19
	19-19	19-19	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-22	19-22
	19-22	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-28	19-28	19-28
	19-29	19-29	19-29	19-29	19-29	20-38	20-38	20-38	20-38	20-38	23-50	23-50	34-13	34-13
	34-13	34-13	34-25	34-25	34-25	34-28	34-28	34-28	34-28	38-45	38-45	38-45	38-45	38-45
	38-57	38-57	38-57	38-61	38-61	38-61	38-61	38-61	38-63	38-63	38-63	39-20	39-20	39-20
	39-22	39-22	39-22	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-39	39-39	39-39
	39-39	39-42	39-42	39-42	39-42	39-46	39-46	39-46	42-38	42-38	42-38	42-38	47-38	47-38
	47-38	47-38	47-38	47-38	51-43	51-43	51-43	51-43	51-43	51-43	51-48	51-48	51-48	51-51
	51-51	51-51	51-51	51-55	51-55	51-55	52-21	52-21	52-21	52-21	52-24	52-24	52-41	52-41
	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-45	52-45	52-45	52-48	52-48	52-48
	52-48	52-53	52-53	52-53	71-37	71-37	71-37	71-37	71-39	71-39	71-39	74-14	74-14	74-21

M\$PUT1	74-21	74-21	74-21	74-21 0	74-24	74-24	74-24	74-24	74-24 0	74-27	74-27	74-27	74-27	74-27 0
	74-32	74-32	74-32	74-32	74-32 0	74-34	74-34	74-34	74-34 0	74-36	74-36	74-36	74-36	74-36 0
	1-C81 0	7-144 0	15-13	15-13	15-13 0	15-13 0	15-19	15-19	15-19	15-19	15-19	15-19 0	15-19 0	15-19 0
	15-19 0	15-19 0	15-21	15-21	15-21	15-21	15-21	15-21 0	15-21 0	15-21 0	15-21 0	15-21 0	15-24	15-24
	15-24 0	15-24 0	15-30	15-30	15-30 0	15-30 0	15-31	15-31	15-31	15-31	15-31	15-31	15-31	15-31
	15-31	15-31 0	15-31 0	15-31 0	15-31 0	15-31 0	15-31 0	15-31 0	15-31 0	15-31 0	15-33	15-33	15-33 0	15-33 0
	15-39	15-39	15-39 0	15-39 0	15-45	15-45	15-45	15-45	15-45 0	15-45 0	15-45 0	15-45 0	15-48	15-48
	15-48 0	15-48 0	18-16	18-16	18-16	18-16 0	18-16 0	18-16 0	18-20	18-20	18-20	18-20	18-20 0	18-20 0
	18-20 0	18-20 0	19-16	19-16	19-16 0	19-16 0	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-17
	19-17	19-17 0	19-17 0	19-17 0	19-17 0	19-17 0	19-17 0	19-17 0	19-17 0	19-17 0	19-19	19-19	19-19 0	19-19 0
	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20 0	19-20 0	19-20 0	19-20 0	19-20 0
	19-20 0	19-20 0	19-20 0	19-20 0	19-22	19-22	19-22 0	19-22 0	19-23	19-23	19-23	19-23	19-23	19-23
	19-23	19-23	19-23	19-23 0	19-23 0	19-23 0	19-23 0	19-23 0	19-23 0	19-23 0	19-23 0	19-23 0	19-28	19-28
	19-28 0	19-28 0	19-29	19-29	19-29	19-29	19-29 0	19-29 0	19-29 0	19-29 0	20-38	20-38	20-38	20-38 0
	20-38 0	20-38 0	23-50	23-50	23-50 0	23-50 0	34-13	34-13	34-13	34-13 0	34-13 0	34-13 0	34-25	34-25
	34-25 0	34-25 0	34-28	34-28	34-28 0	34-28 0	38-45	38-45	38-45	38-45	38-45	38-45 0	38-45 0	38-45 0
	38-45 0	38-45 0	38-57	38-57	38-57 0	38-57 0	38-61	38-61	38-61	38-61 0	38-61 0	38-61 0	38-63	38-63
	38-63 0	38-63 0	39-20	39-20	39-20	39-20 0	39-20 0	39-20 0	39-22	39-22	39-22 0	39-22 0	39-33	39-33
	39-33	39-33	39-33	39-33	39-33	39-33 0	39-33 0	39-33 0	39-33 0	39-33 0	39-33 0	39-33 0	39-39	39-39
	39-39	39-39 0	39-39 0	39-39 0	39-42	39-42	39-42	39-42 0	39-42 0	39-42 0	39-42 0	39-42 0	39-46 0	39-46 0
	42-38	42-38	42-38	42-38 0	42-38 0	42-38 0	47-38	47-38	47-38	47-38	47-38	47-38	47-38 0	47-38 0
	47-38 0	47-38 0	51-43	51-43	51-43	51-43	51-43 0	51-43 0	51-43 0	51-43 0	51-48	51-48	51-48	51-48 0
	51-48 0	51-48 0	51-51	51-51	51-51	51-51 0	51-51 0	51-51 0	51-55	51-55	51-55 0	51-55 0	52-21	52-21
	52-21	52-21 0	52-21 0	52-21 0	52-24	52-24	52-24 0	52-24 0	52-41	52-41	52-41	52-41	52-41	52-41
	52-41	52-41 0	52-41 0	52-41 0	52-41 0	52-41 0	52-41 0	52-41 0	52-45	52-45	52-45	52-45 0	52-45 0	52-45 0
	52-48	52-48	52-48	52-48 0	52-48 0	52-48 0	52-53	52-53	52-53 0	52-53 0	71-37	71-37	71-37 0	71-37 0
	71-39	71-39	71-39 0	71-39 0	74-14	74-14	74-14 0	74-14 0	74-21	74-21	74-21	74-21	74-21 0	74-21 0
	74-21 0	74-21 0	74-24	74-24	74-24	74-24	74-24 0	74-24 0	74-24 0	74-24 0	74-27	74-27	74-27	74-27
	74-27 0	74-27 0	74-27 0	74-27 0	74-32	74-32	74-32	74-32	74-32 0	74-32 0	74-32 0	74-32 0	74-34	74-34
M\$RADI	74-34	74-34 0	74-34 0	74-34 0	74-36	74-36	74-36	74-36	74-36 0	74-36 0	74-36 0	74-36 0	74-36 0	74-36 0
	1-D77 0	7-144 0	23-17	23-17 0	23-23	23-23 0	23-25	23-25 0	23-52	23-52 0	23-56	23-56 0	23-60	23-60 0
	23-64	23-64 0	23-71	23-71 0	23-73	23-73 0	23-81	23-81 0	23-118	23-118 0	71-41	71-41 0	71-54	71-54 0
	71-61	71-61 0	71-70	71-70 0	71-78	71-78 0	71-80	71-80 0	71-85	71-85 0	71-87	71-87 0	71-92	71-92 0
	71-98	71-98 0	71-103	71-103 0	80-55	80-55 0	80-57	80-57 0	80-59	80-59 0	80-61	80-61 0	80-61 0	80-61 0
M\$RBRO	1-C52 0	7-144 0												
M\$RNRO	1-C62 0	7-144 0	35-21	35-21 0										
M\$SETS	1-D32 0	7-144 0	7-170	7-170 0	9-10	9-10 0	10-10	10-10 0	11-51	11-51 0	15-9	15-9 0	15-16	15-16 0
	15-27	15-27 0	15-36	15-36 0	15-42	15-42 0	69-41	69-41 0	69-47	69-47 0	70-8	70-8 0	71-8	71-8 0
	73-10	73-10 0	74-8	74-8 0	75-8	75-8 0	76-9	76-9 0	77-50	77-50 0	77-66	77-66 0	80-43	80-43 0
	80-53	80-53 0	81-12	81-12 0										
M\$STAR	1-A33 0	7-144 0												
M\$SVC	1-C33 0	7-144 0	15-13	15-13 0	15-14	15-14 0	15-19	15-19 0	15-21	15-21 0	15-24	15-24 0	15-25	15-25 0
	15-30	15-30 0	15-31	15-31 0	15-33	15-33 0	15-34	15-34 0	15-39	15-39 0	15-40	15-40 0	15-45	15-45 0
	15-48	15-48 0	15-49	15-49 0	18-16	18-16 0	18-20	18-20 0	19-16	19-16 0	19-17	19-17 0	19-19	19-19 0
	19-20	19-20 0	19-22	19-22 0	19-23	19-23 0	19-28	19-28 0	19-29	19-29 0	20-38	20-38 0	23-17	23-17 0
	23-23	23-23 0	23-25	23-25 0	23-50	23-50 0	23-52	23-52 0	23-56	23-56 0	23-60	23-60 0	23-64	23-64 0
	23-71	23-71 0	23-73	23-73 0	23-81	23-81 0	23-118	23-118 0	27-38	27-42	27-45	27-48	27-51	27-54
	27-57	27-60	27-63	27-66	27-69	27-72	27-75	27-78	27-81	27-84	34-13	34-13 0	34-18	34-25
	34-25 0	34-28	34-28 0	34-34	35-21	35-21 0	38-45	38-45 0	38-57	38-57 0	38-61	38-61 0	38-63	38-63 0
	39-20	39-20 0	39-22	39-22 0	39-33	39-33 0	39-39	39-39 0	39-42	39-42 0	39-46	39-46 0	42-38	42-38 0
	47-38	47-38 0	51-43	51-43 0	51-48	51-48 0	51-51	51-51 0	51-55	51-55 0	52-21	52-21 0	52-24	52-24 0
	52-41	52-41 0	52-45	52-45 0	52-48	52-48 0	52-53	52-53 0	69-61 0	69-76	69-76 0	71-10	71-10 0	71-12
	71-12 0	71-16	71-16 0	71-18	71-18 0	71-22	71-22 0	71-24	71-24 0	71-28	71-28 0	71-37	71-37 0	71-39
	71-39 0	71-41	71-41 0	71-54	71-54 0	71-61	71-61 0	71-70	71-70 0	71-78	71-78 0	71-80	71-80 0	71-85
	71-85 0	71-87	71-87 0	71-92	71-92 0	71-98	71-98 0	71-103	71-103 0	72-10	72-10 0	72-11	72-11 0	72-13
	72-13 0	72-109	72-109 0	72-124	72-124 0	73-18	73-18 0	74-10	74-10 0	74-14	74-14 0	74-17	74-17 0	74-21

	74-21#	74-24	74-24#	74-27	74-27#	74-32	74-32#	74-34	74-34#	74-36	74-36#	74-47	74-47#	74-62
	74-62#	75-19#	75-34	75-34#	76-20#	76-35	76-35#	77-106	77-106#	77-119	77-119#	78-14	78-14#	78-137
	78-137#													
M\$TLAB	1-C29#	7-144#	15-13#	15-14#	15-19#	15-21#	15-24#	15-25#	15-30#	15-31#	15-33#	15-34#	15-39#	15-40#
	15-45#	15-48#	15-49#	18-16#	18-20#	19-16#	19-17#	19-19#	19-20#	19-22#	19-23#	19-28#	19-29#	20-38#
	23-17#	23-23#	23-25#	23-50#	23-52#	23-56#	23-60#	23-64#	23-71#	23-73#	23-81#	23-118#	27-38#	27-42#
	27-45#	27-48#	27-51#	27-54#	27-57#	27-60#	27-63#	27-66#	27-69#	27-72#	27-75#	27-78#	27-81#	27-84#
	34-13#	34-18#	34-25#	34-28#	34-34#	35-21#	38-45#	38-57#	38-61#	38-63#	39-20#	39-22#	39-33#	39-39#
	39-42#	39-46#	42-38#	47-38#	51-43#	51-48#	51-51#	51-55#	52-21#	52-24#	52-41#	52-45#	52-48#	52-53#
	69-76#	71-10#	71-12#	71-16#	71-18#	71-22#	71-24#	71-28#	71-37#	71-39#	71-41#	71-54#	71-61#	71-70#
	71-78#	71-80#	71-85#	71-87#	71-92#	71-98#	71-103#	72-10#	72-11#	72-13#	72-109#	72-124#	73-18#	74-10#
	74-14#	74-17#	74-21#	74-24#	74-27#	74-32#	74-34#	74-36#	74-47#	74-62#	75-34#	76-35#	77-106#	77-119#
	78-14#	78-137#												
M\$TSTL	1-C21#	7-144#	15-13	15-13#	15-14	15-14#	15-19	15-19#	15-21	15-21#	15-24	15-24#	15-25	15-25#
	15-30	15-30#	15-31	15-31#	15-33	15-33#	15-34	15-34#	15-39	15-39#	15-40	15-40#	15-45	15-45#
	15-48	15-48#	15-49	15-49#	18-16	18-16#	18-20	18-20#	19-16	19-16#	19-17	19-17#	19-19	19-19#
	19-20	19-20#	19-22	19-22#	19-23	19-23#	19-28	19-28#	19-29	19-29#	20-38	20-38#	23-17	23-17#
	23-23	23-23#	23-25	23-25#	23-50	23-50#	23-52	23-52#	23-56	23-56#	23-60	23-60#	23-64	23-64#
	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	27-38	27-38#	27-38	27-38#	27-42	27-42#
	27-45	27-45#	27-45#	27-48	27-48#	27-48#	27-51	27-51#	27-51#	27-51#	27-54	27-54#	27-54	27-57
	27-57#	27-60	27-60#	27-60#	27-63	27-63#	27-63#	27-66	27-66#	27-66#	27-69	27-69#	27-69#	27-72
	27-72#	27-72#	27-75	27-75#	27-75#	27-78	27-78#	27-78#	27-81	27-81#	27-81#	27-84	27-84#	27-84#
	34-13	34-13#	34-18	34-18#	34-18#	34-25	34-25#	34-28	34-28#	34-34	34-34#	34-34#	35-21	35-21#
	38-45	38-45#	38-57	38-57#	38-61	38-61#	38-63	38-63#	39-20	39-20#	39-22	39-22#	39-33	39-33#
	39-39	39-39#	39-42	39-42#	39-46	39-46#	42-38	42-38#	47-38	47-38#	51-43	51-43#	51-48	51-48#
	51-51	51-51#	51-55	51-55#	52-21	52-21#	52-24	52-24#	52-41	52-41#	52-45	52-45#	52-48	52-48#
	52-53	52-53#	69-76	69-76#	71-10	71-10#	71-12	71-12#	71-16	71-16#	71-18	71-18#	71-22	71-22#
	71-24	71-24#	71-28	71-28#	71-37	71-37#	71-39	71-39#	71-41	71-41#	71-54	71-54#	71-61	71-61#
	71-70	71-70#	71-78	71-78#	71-80	71-80#	71-85	71-85#	71-87	71-87#	71-92	71-92#	71-98	71-98#
	71-103	71-103#	72-10	72-10#	72-11	72-11#	72-13	72-13#	72-109	72-109#	72-124	72-124#	73-18	73-18#
	74-10	74-10#	74-14	74-14#	74-17	74-17#	74-21	74-21#	74-24	74-24#	74-27	74-27#	74-32	74-32#
	74-34	74-34#	74-36	74-36#	74-47	74-47#	74-62	74-62#	75-34	75-34#	76-35	76-35#	77-106	77-106#
	77-119	77-119#	78-14	78-14#	78-137	78-137#								
M\$WORD	1-C94#	7-144#	7-198	7-198#	8-8	8-8	8-8#	23-17	23-17	23-17#	23-17#	23-23	23-23	23-23#
	23-23#	23-25	23-25	23-25#	23-25#	23-52	23-52	23-52#	23-52#	23-56	23-56	23-56#	23-56#	23-60
	23-60	23-60#	23-60#	23-64	23-64	23-64#	23-64#	23-71	23-71	23-71#	23-71#	23-73	23-73	23-73#
	23-73#	23-81	23-81	23-81#	23-81#	23-118	23-118	23-118#	23-118#	27-38	27-38	27-38	27-38#	27-42
	27-42	27-42	27-42#	27-45	27-45	27-45	27-45#	27-48	27-48	27-48	27-48#	27-51	27-51	27-51
	27-51#	27-54	27-54	27-54#	27-54#	27-57	27-57	27-57	27-57#	27-60	27-60	27-60	27-60#	27-63
	27-63	27-63	27-63#	27-66	27-66	27-66#	27-69	27-69	27-69#	27-69	27-69#	27-72	27-72	27-72
	27-72#	27-75	27-75	27-75#	27-75#	27-78	27-78	27-78	27-78#	27-81	27-81	27-81	27-81#	27-84
	27-84	27-84	27-84#	34-18	34-18	34-18	34-18#	34-34	34-34	34-34	34-34#	34-34#	69-61	69-61#
	71-22#	71-41	71-41	71-41#	71-41#	71-54	71-54	71-54#	71-54#	71-61	71-61	71-61#	71-61#	71-16#
	71-70	71-70#	71-70#	71-78	71-78	71-78#	71-78#	71-80	71-80	71-80#	71-80#	71-85	71-85	71-85#
	71-85#	71-87	71-87	71-87#	71-87#	71-92	71-92	71-92#	71-92#	71-98	71-98	71-98#	71-98#	71-103
	71-103	71-103#	71-103#	72-11#	72-109#	74-17#	74-47#	75-19	75-19#	76-20	76-20#	80-55	80-55#	80-57
	80-57#	80-59	80-59#	80-61	80-61#	82-16	82-16#							
M\$XFER	1-882#	7-144#												
MANUAL	1-;62#	7-144#	77-119											
MEMORY	1-;66#	7-144#												
OPEN	1-;71#	7-144#												
POINTE	1-;76#	7-144#	7-176											
POP	7-112#	20-39	21-33	23-116	24-42	27-32	29-68	31-32	33-24	35-48	36-26	37-26	38-84	39-47
	40-25	42-53	43-51	46-23	47-40	49-48	50-33	50-63	51-58	52-54	58-24	59-21	59-24	59-45
	60-27	61-34	62-49	63-24	63-43	64-21	64-36	65-141	66-26	66-51	66-74	66-99	72-83	
PRINTB	1-;39#	7-144#	15-13	15-19	15-21	15-24	15-30	15-31	15-33	15-39	15-45	15-48	18-16	18-20

