

RP07

FMTR/SCANNER
CZRJKA0

AH-F957A-MC
FICHE 1 OF 1

MAY 1983
COPYRIGHT © 1983
MADE IN USA



The main body of the document is a large, dense grid of data. Each cell in the grid contains a small, structured table or form. The text within these cells is extremely faint and difficult to read, but the overall layout suggests a comprehensive data set or a series of related forms. The grid covers most of the page area below the header.

.REM @

IDENTIFICATION

PRODUCT CODE: AC-F956A-MC
PRODUCT NAME: CZRJKAO RP07 FORMAT/SCANNER
PRODUCT DATE: JANUARY 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 VARIOUS DATA PATTERNS 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 - CHOUS

1.4 DIAGNOSTIC HIERARCY 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 (CHOUS).

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 (O) 176700 ?
VECTOR ADRS (O) 254 ?
BR LEVEL (O) 5 ?
DRIVE # (O) 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).

```
"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).

A '0' OR '2' RESPONSE TO THE PREVIOUS OPTION PROMPT WILL ALSO CAUSE THE FOLLOWING WARNING MESSAGE TO BE PRINTED TO THE OUTPUT DEVICE AND ASK A VERIFY QUESTION.

```
"      ! CUSTOMER DATA WILL BE OVERWRITTEN !  
-----  
CONTINUE (L) ? "
```

THE USER MAY OPT TO STOP ANY FORMATTING OPERATION AT THIS POINT BY ANSWERING 'N' TO THE PREVIOUS QUESTION OR CONTINUE TO THE NEXT QUESTION WITH A 'Y' RESPONSE.

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 D'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 MODIFYING THE TD BY USING MODIFY OPTION '0', 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', TO INSERT A DEFECT WHICH IS IN TD OF OF A PARTICULAR DISK ADDRESS;

NOTE: THE NUMBER OF WORDS FROM INDEX NEEDED TO MOVE A TD SHOULD ALWAYS BE 118.(DECIMAL) AND TO MOVE HDR 0 SHOULD ALWAYS BE 163.(DECIMAL).

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' TO INSERT A DEFECT ON A PARTICULAR DISK ADDRESS. IN THIS EXAMPLE WE WILL ADD A STACKED DEFECT 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 ? 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

BIT 13 0=18 BIT FMT
BIT 14-15 ALWAYS 0
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 (D) ? 8<CR>

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

UNIT 2
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 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.

- 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 ** TD RECORD IS FULL, INPUT DATA REJECTED -

THIS MESSAGE IS GENERATED WHEN A USER ATTEMPTS TO ADD A DEFECT SKIP TO A TRACK DESCRIPTOR RECORD WHICH IS ALREADY FULL (4 DEFECTS).

- ** WARNING ** CYL XXX., TRK YY. HAS 4 DEFECTS IN TD RECORD -

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 ** USR BSF IS FULL, NO MORE ENTRIES ALLOWED -

THIS MESSAGE IS GENERATED WHEN AN ATTEMPT IS MADE TO ADD A BAD SPOT TO A FULL USER BAD SECTOR FILE (DEC144).

- ** WARNING ** CORRUPT BAD SECTOR FILE, 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)
3. VERIFY TRACK FORMAT, WITH WRITE CHECK (IF ENABLED)
4. CREATE-RESTORE DEC STD 144 AREA.

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.
3. READ HEADERS (COMMAND MODIFIER BIT=1) AND CHECK AGAINST MEMORY MAP.

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 APPROXIMATELY 2 HOURS 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)

5. VERIFY TRACK FORMAT, WITH WRITE CHECK

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)
5. VERIFY TRACK FORMAT, WITH WRITE CHECK

@

.REM @

VERSION (CZRJK-A-0)

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

@

1
 2
 135
 137
 163
 165 000000
 166 002000
 168
 170
 171
 172
 173
 174
 176
 193
 197 002000
 002000 103
 002001 132
 002002 122
 002003 112
 002004 113
 002005 000
 002006 000
 002007 000
 002010
 002010 101
 002011
 002011 060
 002012
 002012 000001
 002014
 002014 001000
 002016
 002016 027716
 002020
 002020 000000
 002022
 002022 002130
 002024
 002024 000000
 002026
 002026 117664
 002030
 002030 000000
 002032
 002032 000000
 002034
 002034 000000
 002036
 002036 000000
 002040
 002040 002124
 002042
 002042 000000
 002044
 002044 000000
 002046

```

:*LAST REVISION 01-JAN-83
.TITLE CZRJKAO 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 /A/
L$DEPO::         ;0
      .ASCII /O/
L$UNIT::        ;NUMBER OF UNITS
      .WORD T$PTHV
L$TIML::        ;LONGEST TEST TIME
      .WORD 1000
L$HPCP::        ;POINTER TO H.W. QUES.
      .WORD L$HARD
L$SPCP::        ;POINTER TO S.W. QUES.
      .WORD 0
L$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$DTYP::        ;DIAGNOSTIC TYPE
      .WORD 0
L$APT::         ;APT EXPANSION
      .WORD 0
L$DTP::         ;PTR. TO DISPATCH TABLE
      .WORD L$DISPATCH
L$PRIO::        ;DIAGNOSTIC RUN PRIORITY
      .WORD 0
L$ENVI::        ;FLAGS DESCRIBE HOW IT WAS SETUP
      .WORD 0
L$EXP1::        ;EXPANSION WORD
    
```


002046 000000
 002050
 002050 003
 002051 003
 002052
 002052 000000
 002054 000000
 002056
 002056 000000
 002060
 002060 002600
 002062
 002062 000000
 002064
 002064 000000
 002066
 002066 000000
 002070
 002070 000000
 002072
 002072 000000
 002074
 002074 000000
 002076
 002076 002606
 002100
 002100 104035
 002102
 002102 000000
 002104
 002104 025340
 002106
 002106 026336
 002110
 002110 026334
 002112
 002112 025332
 002114
 002114 000000
 002116
 002116 000000
 002120
 002120 000000

LSMREV:: .WORD 0 ;SVC REV AND EDIT #
 .BYTE CSREVISION
 .BYTE CSEDIT
 LSEF:: .WORD 0 ;DIAG. EVENT FLAGS
 .WORD 0
 LSSPC:: .WORD 0
 L\$DEVP:: .WORD 0 ; POINTER TO DEVICE TYPE LIST
 .WORD LSDVTYP
 L\$REPP:: .WORD 0 ;PTR. TO REPORT CODE
 .WORD 0
 L\$EXP4:: .WORD 0
 .WORD 0
 L\$EXP5:: .WORD 0
 .WORD 0
 L\$AUT:: .WORD 0 ;PTR. TO ADD UNIT CODE
 .WORD 0
 L\$DUT:: .WORD 0 ;PTR. TO DROP UNIT CODE
 .WORD 0
 L\$LUN:: .WORD 0 ;LUN FOR EXERCISERS TO FILL
 .WORD 0
 L\$DESP:: .WORD 0 ;POINTER TO DIAG. DESCRIPTION
 .WORD L\$DESC
 L\$LOAD:: EMT E\$LOAD ;GENERATE SPECIAL AUTOLOAD EMT
 L\$ETP:: .WORD 0 ;POINTER TO ERR_TBL
 .WORD 0
 L\$ICP:: .WORD 0 ;PTR. TO INIT CODE
 .WORD L\$INIT
 L\$CCP:: .WORD 0 ;PTR. TO CLEAN-UP CODE
 .WORD L\$CLEAN
 L\$ACP:: .WORD 0 ;PTR. TO AUTO CODE
 .WORD L\$AUTO
 L\$PRT:: .WORD 0 ;PTR. TO PROTECT TABLE
 .WORD L\$PROT
 L\$TEST:: .WORD 0 ;TEST NUMBER
 .WORD 0
 L\$DLY:: .WORD 0 ;DELAY COUNT
 .WORD 0
 L\$HIME:: .WORD 0 ;PTR. TO HIGH MEM
 .WORD 0

1
2
3
4
5
6
7
8
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.
:--

002122 000001
002124
002124 026660

.WORD 1
L\$DISPATCH:
.WORD T1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
25

.SBTTL DEFAULT HARDWARE P-TABLE

:++
: THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
: THE TEST-DEVICE PARAMETERS. THE STRUCTURE OF THIS TABLE
: IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
: AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
:--

002126 000004
002130
002130
002130 176700
002132 000254
002134 000240
002136 000000
002140

.WORD L10000-L\$HW/2
L\$HW::
DFPTBL::
.WORD 176700 :RPCS1 BASE REGISTER ADDRESS
.WORD 254 :VECTOR ADDRESS
.WORD 240 :BR LEVEL 5 DEVICE
.WORD 0 :DRIVE NUMBER

L10000:

1
2
3
4
5
6
7
8
9
10 002140 000000
002142
002142
11
19
20 002142

.SBTTL SOFTWARE P-TABLE

:++
: THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
: PROGRAM AS OPERATIONAL PARAMETERS. THESE PARAMETERS ARE
: SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
: AT RUN TIME.
:--

.WORD L10001-L\$\$W/2
L\$\$W::
SFPTBL::

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 DIFINITIONS

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
000000

PRI01== 40
PRI00== 0

·
·
· OPERATOR FLAG BITS
·

000004
000010
000020
000040
000100
000200
000400
001000
002000
004000
010000
020000
040000
100000

· EVL== 4
· LOT== 10
· ADR== 20
· IDU== 40
· ISR== 100
· UAM== 200
· BOE== 400
· PNT== 1000
· PRI== 2000
· IXE== 4000
· IBE== 10000
· IER== 20000
· LOE== 40000
· HOE== 100000

RHXX REGISTERS

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
55
56
57

.SBTTL RHXX REGISTERS

:CONTROL AND STATUS REGISTER 1 (RPCS1)

000100	INTEN	== 100	: INTERRUPT ENABLE (BIT #6)
000200	RDY	== 200	: READY (BIT # 7)
000400	A16	== 400	: HIGH ORDER BUS ADDRESS BIT (BIT # 8)
001000	A17	== 1000	: HIGH ORDER BUS ADDRESS BIT (BIT # 9)
002000	PSEL	== 2000	: PORT SELECT (BIT # 10)
020000	MCPE	== 20000	: MASSBUS PARITY ERROR (BIT # 13)
040000	TRE	== 40000	: TRANSFER ERROR (BIT # 14)
100000	SC	== 100000	: SPECIAL CONDITION (BIT # 15)

:WORD COUNT REGISTER (RPWC)
:EACH BIT IS CALLED BY BIT NUMBER

:BUS ADDRESS REGISTER (RPBA)
:EACH BIT IS CALLED BY BIT NUMBER

:CONTROL AND STATUS REGISTER 2 (RPCS2)

000001	US1	== 1	: UNIT SELECT (BIT #0)
000002	US2	== 2	: UNIT SELECT (BIT #1)
000004	US4	== 4	: UNIT SELECT (BIT #2)
000010	BAI	== 10	: BUS ADDRESS INCREMENT INHIBIT (BIT #3)
000020	PAT	== 20	: MASSBUS PARITY INHIBIT (BIT #4)
000040	CLR	== 40	: CLEAR (BIT #5)
000100	IR	== 100	: INPUT READY (BIT #6)
000200	OR	== 200	: OUTPUT READY (BIT #7)
000400	MPE	== 400	: MASSBUS PARITY ERROR (BIT #8)
001000	MXF	== 1000	: MISSED TRANSFER ERROR (BIT #9)
002000	PGE	== 2000	: PROGRAM ERROR (BIT #10)
004000	NEM	== 4000	: NON EXISTENT MEMORY (BIT #11)
010000	NED	== 10000	: NON EXISTENT DRIVE (BIT #12)
020000	UPE	== 20000	: UNIBUS PARITY ERROR
040000	WCE	== 40000	: WRITE CHECK ERROR (BIT #14)
100000	DLT	== 100000	: DATA LATE (BIT #15)

:DATA BUFFER REGISTER (RPDB)
:EACH BIT IS DEFINED BY BIT NUMBER

.SBTTL RP07 REGISTERS

:CONTROL AND STATUS 1 (#00)

000001	GO	== 1	: GO BIT (BIT #0)
000002	F1	== 2	: FUNCTION CODE BIT #1
000004	F2	== 4	: FUNCTION CODE BIT #2
000010	F3	== 10	: FUNCTION CODE BIT #3
000020	F4	== 20	: FUNCTION CODE BIT #4
000040	F5	== 40	: FUNCTION CODE BIT #5
004000	DVA	== 4000	: DEVICE AVAILABLE (BIT #11)

```

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)

```

```

RP07 REGISTERS

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
214      000021      READIN   == 21     :READ IN PRESET
215      000031      SEARCH   == 31     :SEARCH
216      000035      DIAG     == 35     :DIAGNOSTIC MODE
217      000051      WCKD     == 51     :WRITE CHECK DATA
218      000053      WCKHD    == 53     :WRITE CHECK HEADER & DATA
219      000061      WRTDAT   == 61     :WRITE DATA
220      000063      FMTRK    == 63     :FORMAT TRACK
221      000065      WRTTD    == 65     :WRITE TRACK DESCRIPTOR
222      000071      RDDAT    == 71     :READ DATA
223      000073      RDHD     == 73     :READ HEADER & DATA
224      000075      RDTD     == 75     :READ TRACK DESCRIPTOR
225
226      ;SOME TRACK FORMAT EQUATES FOR 16 BIT MODE (IN WORDS)
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      HDRO     == 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
9          002142 002142 BGNPAT ==.
10         002144 030221 .WORD 030221 ;PATTERN #1 (WORST CASE)
11         002146 147556 .WORD 030221 ;PATTERN #2 (NOT WORST CASE)
12         002150 147556 .WORD 147556
13         002152 002152 .WORD 147556
14
16         002152 000000 CONTIN:: .WORD 0 ;CONTINUE TO OVERWRITE CUSTOMER DATA IF = 1
18         002154 000000 WRTFE2:: .WORD 0 ;USED TO PERMIT REWRITE OF THE 2ND FE CYLINDER
19         002156 000000 DRVPAR:: .WORD 0 ;WHEN EQ TO 1 CHANGE DRIVE PARAMETERS
20         002160 000000 OPTION:: .WORD 0 ;OPTIONS; 0=FORMAT, 1=VERIFY, 2=SCAN, 3=MODIFY,
21         ; 4=LIST, 5=WRITE FE-2 AND 6=HELP
22         002162 000000 FORMT:: .WORD 0 ;FORMAT OPTION; ENABLED= 1, DISABLED= 0
23         002164 000000 NOWRCK:: .WORD 0 ;USED TO INHIBIT WRITE CHECK COMMAND
24         002166 000000 VRIFY:: .WORD 0 ;VERIFY OPTION; ENABLED= 1, DISABLED= 0
25         002170 000000 SCANR:: .WORD 0 ;SCAN OPTION; ENABLED= 1, DISABLED= 0
26         002172 000000 ENWTTD:: .WORD 0 ;USED TO ENABLE WRITE TD OPTION AFTER PACK SCAN
27         002174 000000 MODTD:: .WORD 0 ;MODIFY OPTION; ENABLED= 1, DISABLED= 0
28         002176 000000 MODBY:: .WORD 0 ;MODIFY TD BY; 0=WORDS, 1=INDEX, 2=CHANGE
29         002200 000000 LIST:: .WORD 0 ;LIST OPTION; ENABLED= 1, DISABLED= 0
33         002202 000000 LISHDR:: .WORD 0 ;USED TO LIST HEADER INFO IN DEFECT SECTORS
34
35         002204 000000 MINCYL:: .WORD 0 ;USED TO DETERMINE MIN CYLINDER OF OPERATION
36         002206 001166 MAXCYL:: .WORD 630. ;USED TO DETERMINE MAX CYLINDER OF OPERATION
37         002210 000000 MINTRK:: .WORD 0 ;USED TO DETERMINE MIN TRACK OF OPERATION
38         002212 000037 MAXTRK:: .WORD 31. ;USED TO DETERMINE MAX TRACK OF OPERATION
39
40         002214 000000 TEMPA:: .WORD 0 ;USED FOR SOFTWARE CALCULATIONS
41         002216 001166 ENDCYL:: .WORD 630. ;DEFAULT CYLINDER MAX ON DRIVE
42         002220 000037 ENDTRK:: .WORD 31. ;HIGHEST RP07 TRACK ADDRESS
43         002222 000005 ERRMAX:: .WORD 5 ;USED TO CONTROL MAX ERRORS
44         002224 000310 ENDPTR:: .WORD 50.*4 ;USED TO CREATE BUFFER BOUNDARIES
45         002226 000000 ENDTAB:: .WORD 0 ;USED TO MARK BUFFER BOUNDARIES
46         002230 000000 BADDR:: .WORD 0 ;CONTAINS BUS ADDR FOR DRIVER MODULE
47         002232 032250 LASLOC:: .WORD DEFBUF ;USED TO LINK BUFFER TO BUFFER TRANSFERS
48
49         002234 000 CEMODE:: .BYTE 0 ;MODE CONTROL FOR CE CYLINDER
50         002235 000 FIRPAS:: .BYTE 0 ;USED TO CONTROL SCANNER ITERATIONS
51         002236 000 SCANIT:: .BYTE 0 ;# OF SCAN ITERATIONS DURING A SCAN OPERATION
52         002237 000 INTLEV:: .BYTE 0 ;INTERLEAVE MODE INDICATOR
53         002240 000 SAMSEC:: .BYTE 0 ;# DEFECTS IN CURRENT SECTOR
54         002241 000 NOROOM:: .BYTE 0 ;USED WHEN THE BAD SECTOR FILE IS FULL
55         .EVEN
56         002242 000000 SUPRSS:: .WORD 0 ;CONTROL USED TO SUPPRESS ERROR MESSAGES
57         002244 000000 DESTRK:: .WORD 0 ;DESIRED TRACK ADDRESS (IN UPPER BYTE)
58         002246 000000 DEFCNT:: .WORD 0 ;USED TO TRACK THE HDA DEFECT COUNT
59         002250 000000 NEWCNT:: .WORD 0 ;USED TO TRACK NEWLY FOUND DEFECTS
60         002252 000000 TBLPTR:: .WORD 0 ;USED TO FEED AN ADDRESS TO 'GETNEX'
61         002254 000000 ERRFNC:: .WORD 0 ;USED TO STORE THE FUNCTION AT TIME OF FAILURE
62         002256 000000 SOFSW:: .WORD 0 ;USED FOR SOFTWARE SCRATCH
    
```


GLOBAL DATA SECTION

```

63 002260 000000 ERTTL:: .WORD 0 ;CONTAINS THE TOTAL # OF ERRORS DETECTED
64 002262 000000 ERRCTL:: .WORD 0 ;CONTAINS THE TOTAL # OF DEVICE RETRIES
65 002264 000000 ERRMSK:: .WORD 0 ;ERROR MASK
66 002266 000000 RTYCNT:: .WORD 0 ;USED TO MAINTAIN THE RETRY COUNTER
67 002270 000000 EXPCTD:: .WORD 0 ;USED TO IDENTIFY EXPECTED DATA
68 002272 000000 RECVD:: .WORD 0 ;USED TO IDENTIFY RECEIVED DATA
69 002274 000000 DEF1:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH ONE DEFECT
70 002276 000000 DEF2:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH TWO DEFECTS
71 002300 000000 DEF3:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH THREE DEFECTS
72 002302 000000 DEF4:: .WORD 0 ;ACCUMULATOR FOR TRACKS WITH FOUR DEFECTS
73 002304 000000 DESCYL:: .WORD 0 ;DESIRED CYLINDER ADDRESS
74
75 002306 061 SEC50:: .BYTE 49. ;50 SECTORS (0-49.)
76 002307 031 000 PLTRK:: .BYTE 25.,0 ;INTERLEAVED FILE FOR 16-BIT FORMAT
77 .EVEN
78 002312 002142 PATRN:: .WORD BGNPAT ;POINTER TO BEGINNING OF DATA PATTERNS
79 002314 000006 TDBCNT:: .WORD 6 ;TD BYTE COUNT
80 002316 000110 TDVALU:: .WORD G1+TD ;GAP 1 + TD CONSTANT
81 002320 000045 .WORD G2 ;GAP 2 CONSTANT
82
83 ;*****
84 ; THE FOLLOWING TABLE REPRESENTS THE VARIOUS PARTITIONS WITHIN A SECTOR FOR AN
85 ; RP07. (IN 16 BIT MODE) THE TABLE STRUCTURE MUST NOT BE ALTERED IN ANYWAY.
86 ;
87 ; >>>DO NOT DISTURB THE FOLLOWING TABLE UNDER ANY CIRCUMSTANCES<<<
88 ;
89 002322 000010 PARTBL:: .WORD HDR ;SECTOR HEADER,
90 002324 000040 .WORD G3 ;GAP 3,
91 002326 000022 .WORD MSEG ;LEADING MIN. DATA SEGMENT,
92 002330 000336 .WORD 222. ;DATA FIELD SEGMENT,
93 002332 000022 .WORD MSEG ;TRAILING MIN. DATA SEGMENT,
94 002334 000045 .WORD G4 ;GAP 4.
95 ;*****
96 ; THE TWO TD LIMIT WORDS, 'TDLMTS' AND 'TDLMTS+2' ARE REFERENCED FROM
97 ; INDEX TO THE CENTER OF THE FIRST DEFECT SKIP (DS1). WHILE THE TD LIMIT
98 ; WORD 'TDLMTS+4' IS REFERENCED FROM THE CENTER OF THE FIRST DEFECT SKIP (DS1)
99 ; TO THE CENTER OF THE SECOND DEFECT SKIP (DS2). THESE NUMBERS WILL BE USED
100 ; WHEN THE PROGRAM IS TRYING TO DETERMINE IF A TD AND/OR HDR 0 SHOULD
101 ; BE MOVED.
102
103 002336 000166 TDLMTS:: .WORD G1+<DS/2> ;TD MOVED LIMIT
104 002340 000243 .WORD G1+TD+G2+<DS/2> ;HDR 0 MOVED LIMIT
105 002342 000231 .WORD <DS/2>+TD+G2+<DS/2> ;TD & HDR 0 MOVED LIMIT
106 ;*****
107
108 002344 000000 TEXT:: .WORD 0 ;POINTS TO ADDRESS OF TEXT TO BE TYPED
109 002346 000000 CONTLT:: .WORD 0 ;ADDRESS OF CONTROL 'T' SERVICE BUFFER
110 002350 000154 TKWCNT:: .WORD 6*50. ;WORDS FOR TRACK FORMAT (6 WORDS X 50. SECTORS)
111 002352 00J000 TEMP1:: .WORD 0 ;USED TO IDENTIFY SCRATCHES
112 002354 000000 TEMP2:: .WORD 0 ;USED TO MEASURE SCRATCH LENGTH
113 002356 000000 HICYL:: .WORD 0 ;USED TO MAP THE HI ADDRESS OF A SCRATCH
114 002360 000000 LOCYL:: .WORD 0 ;USED TO MAP THE LO ADDRESS OF A SCRATCH
115 002362 000000 NEGWRD:: .WORD 0 ;NEGATED WORD COUNT FOR DRIVER
116 002364 000000 ECCWRD:: .WORD 0 ;ECC DATA FOR CREATION OF NEW TD
117 002366 000000 SECADD:: .WORD 0 ;SECTOR ADDRESS OF NEW TD DEFECT
118 002370 000000 FUNCTN:: .WORD 0 ;FUNCTION COMMAND FOR RP07 DRIVER MODULE
119 002372 000000 DEFSEC:: .WORD 0 ;CONTAINS # OF DEFECTS IN A SECTOR

```

120	002374	000000		DEFTRK:: .WORD	0	:CONTAINS # OF DEFECTS ON A TRACK
121	002376	000000		MINSEG:: .WORD	0	:MINIMUM VALUE FOR A DATA SEGMENT
122	002400	000000		MAXSEG:: .WORD	0	:MAXIMUM VALUE FOR A DATA SEGMENT
123	002402	000000		MORETD:: .WORD	0	:IF EQ 1, THEN MORE TD'S TO MODIFY
124						
125	002404	000000		UNIT:: .WORD	0	:USED TO SELECT A UNIT NUMBER
126	002406	176700		RPADR:: .WORD	176700	:CONTAINS RPCS1 BASE ADDRESS
127	002410	000254	000240	RPVEC:: .WORD	254,5*32.	:CONTAINS VECTOR ADDRESS & BR LEVEL
128	002414	000050		RHEXT:: .WORD	50	:CONTAINS RH70 OFFSET TO RPBAE
129	002416	000000		RHTYPE:: .WORD	0	:CONTAINS RHXX TYPE; RH11= 0, RH70= 1
130	002420	000000		DRVNO:: .WORD	0	:DRIVE NUMBER
131	002422	000000		DRVSN:: .WORD	0	:STORAGE FOR EACH S/N DIGIT
132						
133	002424	176700		RPCS1:: .WORD	176700	:BASE ADDRESS USED FOR THE DRIVE
134	002426	176702		RPWC:: .WORD	176702	:WORD COUNT REGISTER
135	002430	176704		RPBA:: .WORD	176704	:BYTE ADDRESS REGISTER
136	002432	176706		RPDA:: .WORD	176706	:DESIRED SECTOR/TRACK ADDRESS
137	002434	176710		RPCS2:: .WORD	176710	:RP07 STATUS REGISTER
138	002436	176712		RPDS:: .WORD	176712	:RP07 DRIVE STATUS
139	002440	176714		RPER1:: .WORD	176714	:RP07 ERROR REGISTER #1
140	002442	176716		RPAS:: .WORD	176716	:RP07 ATTENTION SUMMARY PSEUDO REGISTER
141	002444	176720		RPLA:: .WORD	176720	:RP07 LOOK AHEAD REGISTER
142	002446	176722		RPDB:: .WORD	176722	:RP07 DATA BUFFER
143	002450	176724		RPMR1:: .WORD	176724	:RP07 MAINTENANCE REGISTER #1
144	002452	176726		RPDT:: .WORD	176726	:DRIVE TYPE REGISTER
145	002454	176730		RPSN:: .WORD	176730	:RP07 SERIAL NUMBER
146	002456	176732		RPOF:: .WORD	176732	:RP07 OFFSET REGISTER
147	002460	176734		RPDC:: .WORD	176734	:RP07 DESIRED CYLINDER
148	002462	176736		RPCC:: .WORD	176736	:RP07 CURRENT CYLINDER
149	002464	176740		RPER2:: .WORD	176740	:RP07 ERROR REGISTER #2
150	002466	176742		RPER3:: .WORD	176742	:RP07 ERROR REGISTER #3
151	002470	176744		RPEC1:: .WORD	176744	:RP07 ERROR-POSITION
152	002472	176746		RPEC2:: .WORD	176746	:RP07 ERROR PATTERN
153	002474	176750		RPBAE:: .WORD	176750	:RH70 REGISTER
154	002476	176752		RPCS3:: .WORD	176752	:RH70 REGISTER
155						
156				: STORAGE FOR DEVICE REGISTERS		
157						
158	002500			REG:: .BLKW	22.	:BUFFER TO SAVE REGISTERS AFTER AN ERROR
159						
160	002554			DELTA:: .BLKW	4	:BUFFER USED TO STORE DELTA ADJUSTMENTS
161	002564			CMDQUE:: .BLKW	6	:USED FOR A COMMAND SEQUENCE QUEUE
162						

GLOBAL TEXT SECTION

```

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     :
13     :
14     :
15     :
16     :
17     :L$DVTYP::
18     :   .ASCIZ /RP07/
19     :   .EVEN
20
21     : TEST DESCRIPTION
22     :
23     :
24     :
25     :
26     :
27     :L$DESC::
28     :   .ASCIZ /RP07 FORMAT-VERIFY-SCANNER/
29     :   .EVEN
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     :
55     :
56     :
57     :
58     :
59     :
60     :
61     :
62     :
63     :
64     :
65     :
66     :
67     :
68     :
69     :
70     :

```

000 CRLF:: .ASCIZ /%N/
 000 FRMT00:: .ASCIZ /%T/
 045 FRMT01:: .ASCIZ /%N%ADRIIVE RPCS1 RPWC RPBA RPDA RPCS2 RPDS/
 045 FRMT02:: .ASCIZ /%N%ARPER1 RPAS RPLA RPDB RPMR1 RPDT RPSN/
 045 FRMT03:: .ASCIZ /%N%ARPER2 RPAS RPLA RPDB RPMR1 RPDT RPSN/
 045 FRMT04:: .ASCIZ /%N%ARPOF RPDC RPCC RPER2 RPER3 RPEC1 RPEC2/
 045 FRMT05:: .ASCIZ /%N%ARPOF RPDC RPCC RPER2 RPER3 RPEC1 RPEC2/
 045 FRMT06:: .ASCIZ /%ARPBAE RPCS3/
 045 FRMT07:: .ASCIZ /%N%06%A %06%N/
 045 FRMT10:: .ASCIZ /%N%ADRIIVE %01/
 045 FRMT11:: .ASCIZ /%N%ADONE, RETRIES MADE= %D5%A. ERRORS DETECTED= %D5%A.%N/
 124 FRMT12:: .ASCIZ /%ATRACKS WITH %D1%A DEFECTS= %D5%A.%N/
 045 FRMT13:: .ASCIZ /%N%ATOTAL DEFECTS FOUND= %D5%A./
 065 FRMT14:: .ASCIZ /%D5%A. /
 066 FRMT15:: .ASCIZ /%06%A /
 045 FRMT16:: .ASCIZ /%N%ADRIIVE WORD#1 WORD#2 WORD#3 WORD#4 WORD#5 WORD#6/
 045 FRMT17:: .ASCIZ /%N%06%A %06%A %06%A %06%A %06%A %06%A %06/
 045 FRMT20:: .ASCIZ /%N%ACYL:%D3%A. TRK:%D2%A. /
 122 FRMT21:: .ASCIZ /%ARETRIES MADE= %D5%A./
 120 FRMT22:: .ASCIZ /%APRSNT FUNCT: %T/
 045 FRMT23:: .ASCIZ /%T%A %06%A %06/
 045 FRMT24:: .ASCIZ /%N%ATYPE <CR> TO INPUT (0=140000) DATA:/
 054 FRMT30:: .ASCIZ /%A, NON-INTERLEAVED%N/
 054 FRMT31:: .ASCIZ /%A, INTERLEAVED%N/
 045 FRMT32:: .ASCIZ /%N%ANEW DEFECT(S) DURING SCAN ITERATION %D3%A., TD(S) FOLLOW:/
 045 FRMT33:: .ASCIZ /%N%AEXPCTD:%06%A RECVD:%06/
 045 FRMT35:: .ASCIZ /%N%AWORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5 WORD#6%N/
 066 FRMT36:: .ASCIZ /%06%A (%D3%A.) %06%A (%D2%A.)(%03%A) /

```

71 004600 045 101 040 FRMT37:: .ASCIZ /%A %06% ( %D2%A.) /
72
73 004647 045 116 045 FRMT50:: .ASCII /%N%AOPTIONS FOLLOW;/
74 004672 045 116 045 .ASCII /%N%AO=FORMAT/
75 004706 045 116 045 .ASCII /%N%A1=VERIFY/
76 004722 045 116 045 .ASCII /%N%A2=SCAN/
77 004734 045 116 045 .ASCII /%N%A3=LIST/
78 004746 045 116 045 .ASCII /%N%A4=MODIFY/
79 004762 045 116 045 .ASCIZ /%N%A5=WRITE FE-2%N/
80
81 005005 045 116 045 FRMT60:: .ASCIZ /%N%T%ATD RECORD IS FULL, INPUT DATA REJECTED%N/
82 005064 045 116 045 FRMT61:: .ASCIZ /%N%T%ACYL %D3%A. TRK %D2%A. HAS 4 DEFECTS IN TD RECORD%N/
83 005156 045 116 045 FRMT62:: .ASCIZ /%N%T%ATHERE ARE %D5%A. TRACK(S) WITH 4 DEFECTS%N/
84 005237 045 116 045 FRMT63:: .ASCIZ /%N%T%AUSR BSF BUFFER IS FULL, NO MORE ENTRIES ALLOWED%N/
85 005327 045 116 045 FRMT64:: .ASCIZ /%N%T%ACORRUPT BAD SECTOR FILE, WILL BE INITIALIZED%N/
86
87 .SBTTL GLOBAL ASCII MESSAGE SECTION
88
89 005414 127 122 111 MSWRD:: .ASCIZ /WRITE TD/
90 005425 122 105 101 MSRDTD:: .ASCIZ /READ TD/
91 005435 127 122 124 WRDAT:: .ASCIZ /WRT DATA/
92 005446 127 122 124 WCKDAT:: .ASCIZ /WRT CHK DATA/
93 005463 127 122 124 WCKHDR:: .ASCIZ /WRT CHK HEADER/
94 005502 122 105 103 RECALI:: .ASCIZ /RECAL/
95 005510 106 117 122 FORMAT:: .ASCIZ /FORMAT TRK/
96 005523 122 105 101 RHDATA:: .ASCIZ /READ HEADER/
97 005537 124 105 123 MOLINE:: .ASCIZ /TEST MOL= 1/
98 005553 124 105 123 WLOCK:: .ASCIZ /TEST WRL= 0/
99 005567 124 105 123 DRVRDY:: .ASCIZ /TEST RPDS: BIT7= 1/
100 005612 015 012 105 EXPTD:: .ASCIZ <CR><LF>/EXPCTD: /
101 005625 015 012 122 RCVED:: .ASCIZ <CR><LF>/RCVED: /
102 005640 015 012 103 CURENT:: .ASCIZ <CR><LF>/CURRENT TD,/
103 005656 015 012 124 TDDEF:: .ASCIZ <CR><LF>/TD(S) WITH DEFECTS FOLLOW:/
104 005713 015 012 124 ANDHDR:: .ASCIZ <CR><LF>/TD(S) WITH DEFECTS (INCLUDING HEADER INFO) FOLLOW:/
105 006000 015 012 103 CHANGE:: .ASCIZ <CR><LF>/CHANGED TD,/
106 006016 052 052 040 WARN:: .ASCIZ /** WARNING ** /<BELL><BELL>
107
108 006037 103 110 101 MSG18:: .ASCIZ /CHANGE DRIVE PARAMETERS/
109 006067 105 116 124 MSG7:: .ASCIZ /ENTER OPTION (6=HELP)/
110 006115 115 111 116 MSG13:: .ASCIZ /MIN TRK/
111 006125 115 101 130 MSG14:: .ASCIZ /MAX TRK/
112 006135 115 111 116 MSG15:: .ASCIZ /MIN CYL/
113 006145 115 101 130 MSG16:: .ASCIZ /MAX CYL/
115 006155 007 011 041 MSG25:: .ASCII <BELL>/ ! CUSTOMER DATA WILL BE OVERWRITTEN !/<CR><LF>
116 006226 007 011 055 .ASCII <BELL>/ -----/<CR><LF>
117 006277 103 117 116 .ASCIZ /CONTINUE/
119 006310 111 116 110 MSG17:: .ASCIZ /INHIBIT WRITE CHECK/
120 006334 115 117 104 MSG22:: .ASCIZ /MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE)/
124 006403 104 117 040 MSG19:: .ASCIZ /DO YOU WANT TO RE-WRITE TD(S) WITH NEW DEFECTS/
125 006462 114 111 123 MSG23:: .ASCIZ /LIST HEADER INFO IN DEFECT SECTOR(S)/
126
127 006527 124 104 040 EM1:: .ASCIZ /TD ADDRESS INCORRECT/
128 006554 124 104 040 EM2:: .ASCIZ /TD FORMAT INCORRECT/
129 006600 103 117 115 EM3:: .ASCIZ /COMPOSITE ERROR SET/
130 006624 104 122 111 EM4:: .ASCIZ /DRIVE HUNG, DRY NOT SET IN TIME/
131 006664 104 122 111 EM5:: .ASCIZ /DRIVE WRITE LOCKED/
132 006707 104 122 111 EM6:: .ASCIZ /DRIVE OFFLINE/
    
```


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

.SBTTL GLOBAL ERROR REPORT SECTION

```

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

ERR0::

```

JSR PC,TYPLOC ;REPORT THE ADDRESS OF THE FAILURE
JSR PC,DMPREG ;DUMP THE ERROR REGISTERS
;CR-LF
MOV #CRLF,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
L10002: TRAP C$MSG
    
```

ERR1::

```

JSR PC,TYPLOC ;REPORT THE ADDRESS OF THE FAILURE
;PRINT 'EXPCTD: XXXXXX XXXXXX'
MOV TDCPY2,-(SP)
MOV TDCPY1,-(SP)
MOV #EXPTD,-(SP)
MOV #FRMT23,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #12,SP
    
```

;PRINT 'RECVED: XXXXXX XXXXXX'

```

MOV TDWRD2,-(SP)
MOV TDWRD1,-(SP)
MOV #RCVED,-(SP)
MOV #FRMT23,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #12,SP
JSR PC,DMPREG
    
```

;DUMP THE ERROR REGISTERS
;CR-LF

L10003:

```

MOV #CRLF,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
ADD #4,SP
TRAP C$MSG
    
```

ERR2::

```

JSR PC,TYPLOC ;REPORT THE ADDRESS OF THE FAILURE
;PRINT 'DRIVE WORD#1 WORD#2 WORD#3 WORD#4 WORD#5 WORD#6'
MOV #FRMT16,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTB
    
```

```

007406
007406 004737 010472
007412 004737 010556
007416 012746 002642
007422 012746 000001
007426 010600
007430 104414
007432 062706 000004
007436 104423
007440
007440 004737 010472
007444 013746 030216
007450 013746 030214
007454 012746 005612
007460 012746 004125
007464 012746 000004
007470 010600
007472 104414
007474 062706 000012
007500 013746 030202
007504 013746 030200
007510 012746 005625
007514 012746 004125
007520 012746 000004
007524 010600
007526 104414
007530 062706 000012
007534 004737 010556
007540 012746 002642
007544 012746 000001
007550 010600
007552 104414
007554 062706 000004
007560 104423
007562
007562 004737 010472
007566 012746 003645
007572 012746 000001
007576 010600
007600 104414
    
```


31	007602	062706	000004	ADD	#4,SP	
	007606	013746	030212	MOV	TDWRD6,-(SP)	
	007612	013746	030210	MOV	TDWRD5,-(SP)	
	007616	013746	030206	MOV	TDWRD4,-(SP)	
	007622	013746	030204	MOV	TDWRD3,-(SP)	
	007626	013746	030202	MOV	TDWRD2,-(SP)	
	007632	013746	030200	MOV	TDWRD1,-(SP)	
	007636	013746	002420	MOV	DRVNO,-(SP)	
	007642	012746	003740	MOV	#FRMT17,-(SP)	
	007646	012746	000010	MOV	#10,-(SP)	
	007652	010600		MOV	SP,RO	
	007654	104414		TRAP	C\$PNTB	
	007656	062706	000022	ADD	#22,SP	
32						:CR-LF
33	007662	012746	002642	MOV	#CRLF,-(SP)	
	007666	012746	000001	MOV	#1,-(SP)	
	007672	010600		MOV	SP,RO	
	007674	104414		TRAP	C\$PNTB	
	007676	062706	000004	ADD	#4,SP	
34	007702			L10004:	TRAP	C\$' ;
	007702	104423				
35						
36	007704			ERR3::		
37	007704	004737	010472	JSR	P IYPLOC	:REPORT THE ADDRESS OF THE FAILURE
38						:CR-LF
39	007710	012746	002642	MOV	#CRLF,-(SP)	
	007714	012746	000001	MOV	#1,-(SP)	
	007720	010600		MOV	SP,RO	
	007722	104414		TRAP	C\$PNTB	
	007724	062706	000004	ADD	#4,SP	
40	007730			L10005:	TRAP	C\$MSG
	007730	104423				
41						
42	007732			ERR5::		
43	007732	004737	010472	JSR	PC,TYPLCC	:REPORT THE ADDRESS OF THE FAILURE
44						:PRINT 'EXPCTD: XXXXXX RECVD: XXXXXX'
45	007736	013746	002272	MOV	RECVD,-(SP)	
	007742	013746	002270	MOV	EXPCTD,-(SP)	
	007746	012746	004364	MOV	#FRMT33,-(SP)	
	007752	012746	000003	MOV	#3,-(SP)	
	007756	010600		MOV	SP,RO	
	007760	104414		TRAP	C\$PNTB	
	007762	062706	000010	ADD	#10,SP	
46	007766	004737	010556	JSR	PC,DMPREG	:DUMP THE ERROR REGISTERS
47						:CR-LF
48	007772	012746	002642	MOV	#CRLF,-(SP)	
	007776	012746	000001	MOV	#1,-(SP)	
	010002	010600		MOV	SP,RO	
	010004	104414		TRAP	C\$PNTB	
	010006	062706	000004	ADD	#4,SP	
49	010012			L10006:	TRAP	C\$MSG
	010012	104423				
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      ;
7      ;       JSR      PC,SIZE70      ;CALL ROUTINE
8      ;
9      ;R5 MUST CONTAIN POINTER TO NEW RPCS1 BASE ADDRESS
11     SIZE70: CLR      RHEXT          ;CLEAR RPBAE OFFSET
12     CLR      RHTYPE          ;CLEAR RHXX TYPE REGISTER (RH11)
13     MOV      ERRVEC,-(SP)    ;SAVE CONTENTS OF ERROR VECTOR
14     MOV      #2$,ERRVEC     ;SETUP 'TRAP' RETURN ADDRESS
15     MOV      (R5),R0        ;GET RPCS1 ADDRESS
16     ADD      #50,R0         ;GET REGISTER OFFSET FOR RH70
17     MOV      #10.,R2       ;GET NUMBER OF REGISTERS TO CHECK
18     TST      (R0)+         ;TRAP IF NOT A VALID RPBAE
19     TST      (R0)+         ;TRAP IF NOT A VALID RPCS3
20     MOV      #50,RHEXT     ;LOAD OFFSET FOR RPBAE (22 REGISTER RH)
21     TST      (R0)+         ;TRAP IF NOT A VALID REGISTER
22     DEC      R2            ;DONE WITH ALL 32 REGISTERS ?
23     BNE      1$           ;BR IF NO
24     MOV      #74,RHEXT     ;LOAD OFFSET FOR RPBAE (32 REGISTER RH)
25     BR      3$
26     MOV      #3$, (SP)    ;SETUP RETURN ADDRESS
27     RTI
28
29     3$:  MOV      (R5),R0    ;GET RPCS1 REGISTER
30     MOV      RHEXT,R2     ;GET RPBAE REGISTER OFFSET
31     BEQ      4$           ;BR IF NONE
32     ADD      R0,R2        ;GET RPBAE REGISTER
33     BIS      #A17!A16,(R0) ;SET EXTENDED ADDRESS BITS IN RPCS1
34     CMP      #3,(R2)     ;ARE THE EXTENDED BITS SET IN RPBAE ?
35     BNE      4$           ;BR IF NO
36     CLR      (R2)        ;CLEAR EXTENDED ADDRESS BITS IN RPBAE
37     MOV      (R0),-(SP)   ;SAVE RPCS1 REG CONTENTS
38     BIC      #^C<A17!A16>,(SP)+ ;ARE THE EXTEND BITS CLEAR IN RPCS1 ?
39     BNE      4$           ;BR IF NO
40     INC      RHTYPE       ;SET RHXX TYPE REGISTER (RH70)
41     MOV      (SP)+,ERRVEC ;RESTORE CONTENTS OF ERROR VECTOR
42     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, BADDR, 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     :BADDR  = BUS ADDRESS FOR DATA TRANSFERS.
17     :FUNCTN = COMMAND (FUNCTION) TO BE EXECUTED.
18     :
19     :OUTPUTS ARE:  RPWC, RPDA, RPCS1, RPBA, ERRMSK
20     :
21     :CALL
22     :              JSR      PC,DRIVER      ;SEND COMMAND TO CONTROLLER
23     :*****
24
25 010156 105777 172254  READY:  TSTB  @RPDS      ;IS DRIVE READY SET ?
26 010162 100375          BPL      READY      ;BR IF NO
27 010164 000207          RTS      PC
28
29 010166 005037 002264  DRIVER:  CLR      ERRMSK    ;RESET NO ERROR STATUS
30 010172 005437 002362          NEG      NEGWRD    ;NEGATE THE WORD COUNT ONLY ONCE!
31 010176 032777 040000 172232 1$:  BIT      #ERR,@RPDS  ;DO WE HAVE A COMPOSITE ERROR?
32 010204 001404          BEQ      2$      ;NO, SO SKIP NEXT
33 010206 012737 000002 002264  MOV      #2,ERRMSK  ;LOG 'COMPOSITE ERROR' STATUS
34 010214 000510          BR       8$      ;AND RETURN
35
36 010216 004737 010156 2$:  JSR      PC,READY    ;WAIT FOR DRIVE READY
37 010222 013777 002362 172176  MOV      NEGWRD,@RPWC ;WORD COUNT ---> RP REGISTER
38 010230 013777 002304 172222  MOV      DESCYL,@RPDC ;GET THE CYLINDER ADDRESS TO THE DEVICE REGISTER
39 010236 013777 002244 172166  MOV      DESTRK,@RPDA ;TRACK/SEC ---> RP REGISTER
40 010244 013777 002230 172156  MOV      BADDR,@RPBA  ;LOAD THE BUS ADDRESS REGISTER
41 010252 042777 100000 172170  BIC      #DMD,@RPMR1  ;ASSUME NOT DIAGNOSTIC MODE
42 010260 023727 002304 001166  CMP      DESCYL,#630. ;FE CYLINDER?
43 010266 103403          BLO      3$      ;IF NOT, SKIP NEXT INSTRUCTION
44 010270 052777 100000 172152  BIS      #DMD,@RPMR1  ;SET DIAGNOSTIC MODE
45 010276 013777 002370 172120 3$:  MOV      FUNCTN,@RPCS1 ;GET THE COMMAND TO EXECUTE
46 010304 017746 172114 4$:  MOV      @RPCS1,-(SP) ;GET STATUS
47 010310 042716 177576          BIC      #^C<RDY!GO>,(SP)
48 010314 022726 000200          CMP      #RDY,(SP)+  ;RDY=1,GO=0 ?
49 010320 001371          BNE      4$      ;BR IF NO
50
51     ;FUNCTION COMPLETE, LOOK FOR ERRORS
52
53 010322 032777 040000 172074  BIT      #TRE,@RPCS1  ;IS TRE SET ?
54 010330 001404          BEQ      5$      ;BR IF NO
55 010332 012737 000014 002264  MOV      #14,ERRMSK  ;LOG 'TRANSFER ERROR' STATUS
56 010340 000441          BR       9$
57

```

58	010342	023727	002370	000035	5\$:	CMP	FUNCTN,#DIAG	:WAS IT A "HOUSEKEEP" COMMAND ?
59	010350	101412				BLOS	6\$:BR IF NO
60	010352	005777	172060			TST	@RPDS	:DID WE GET AN UNEXPECTED ATA ?
61	010356	100423				BMI	7\$:BR IF YES
62	010360	005777	172040			TST	@RPCS1	:IS SPECIAL CONDITION SET ?
63	010364	100004				BPL	6\$:BR IF NO
64	010366	012737	000015	002264		MOV	#15,ERRMSK	:LOG 'SPECIAL CONDITION ERROR' STATUS
65	010374	000423				BR	9\$	
66								
67	010376	032777	040000	172032	6\$:	BIT	#ERR,@RPDS	:DID WE GET A COMPOSITE ERROR?
68	010404	001420				BEQ	10\$:BR IF NO
69	010406	032777	040000	172020		BIT	#WCE,@RPCS2	:IS WCE SET ?
70	010414	001410				BEQ	8\$:BR IF NO
71	010416	012737	000007	002264		MOV	#7,ERRMSK	:LOG 'WRITE CHECK ERROR' STATUS
72	010424	000407				BR	9\$	
73								
74	010426	012737	000006	002264	7\$:	MOV	#6,ERRMSK	:LOG 'UNEXPECTED ATTENTION ERROR' STATUS
75	010434	000403				BR	9\$	
76								
77	010436	012737	000002	002264	8\$:	MOV	#2,ERRMSK	:LOG 'COMPOSITE ERROR' STATUS
78	010444	000207			9\$:	RTS	PC	:EXIT
79								
80	010446	005737	002164		10\$:	TST	NOWRCK	:INHIBIT WRITE CHECK OPERATION ?
81	010452	003374				BGT	9\$:BR IF YES
82	010454	022737	000063	002370		CMP	#FMTRK,FUNCTN	:WAS THIS LAST FUNCTION A FORMAT TRACK ?
83	010462	001370				BNE	9\$:BR IF NO
84	010464	004737	011212			JSR	PC,WRITCK	:DO THE WRITE-CHECK OPERATION
85	010470	000636				BR	DRIVER	

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

```
*****  
: 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  
:*****
```

```
010472  
010472 013746 002254  
010476 012746 004103  
010502 012746 000002  
010506 010600  
010510 104414  
010512 062706 000006  
  
010516 113737 002245 002346  
  
010524 013746 002346  
010530 013746 002304  
010534 012746 004020  
010540 012746 000003  
010544 010600  
010546 104414  
010550 062706 000010  
010554 000207
```

```
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
```

```
*****  
: 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  
:*****
```

```
010556  
010556 012746 002650  
010562 012746 000001  
010566 010600  
010570 104415  
010572 062706 000004  
010576 013746 002512  
010602 013746 002510  
010606 013746 002506  
010612 013746 002504  
010616 013746 002502
```

```
DMPREG: ;PRINT 'DRIVE  RPCS1  RPWC  RPBA  RPDA  RPCS2  RPDS'  
MOV #FRMT00,-(SP)  
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)
```

DISK DRIVER

	010622	013746	002500	MOV	REG,-(SP)	
	010626	013746	002420	MOV	DRVNO,-(SP)	
	010632	012746	002741	MOV	#FRMT01,-(SP)	
	010636	012746	000010	MOV	#10,-(SP)	
	010642	010600		MOV	SP,R0	
	010644	104415		TRAP	C\$PNTX	
	010646	062706	000022	ADD	#22,SP	
39						
40	010652	012746	003021	MOV	#FRMT02,-(SP)	:PRINT 'RPER1 RPAS RPLA RPDB RPMR1 RPDT RPSN'
	010656	012746	000001	MOV	#1,-(SP)	
	010662	010600		MOV	SP,R0	
	010664	104415		TRAP	C\$PNTX	
	010666	062706	000004	ADD	#4,SP	
41	010672	013746	002530	MOV	REG+30,-(SP)	
	010676	013746	002526	MOV	REG+26,-(SP)	
	010702	013746	002524	MOV	REG+24,-(SP)	
	010706	013746	002522	MOV	REG+22,-(SP)	
	010712	013746	002520	MOV	REG+20,-(SP)	
	010716	013746	002516	MOV	REG+16,-(SP)	
	010722	013746	002514	MOV	REG+14,-(SP)	
	010726	012746	003112	MOV	#FRMT03,-(SP)	
	010732	012746	000010	MOV	#10,-(SP)	
	010736	010600		MOV	SP,R0	
	010740	104415		TRAP	C\$PNTX	
	010742	062706	000022	ADD	#22,SP	
42						
43	010746	012746	003172	MOV	#FRMT04,-(SP)	:PRINT 'RPOF RPDC RPCC RPER2 RPER3 RPEC1 RPEC2'
	010752	012746	000001	MOV	#1,-(SP)	
	010756	010600		MOV	SP,R0	
	010760	104415		TRAP	C\$PNTX	
	010762	062706	000004	ADD	#4,SP	
44	010766	013746	002546	MOV	REG+46,-(SP)	
	010772	013746	002544	MOV	REG+44,-(SP)	
	010776	013746	002542	MOV	REG+42,-(SP)	
	011002	013746	002540	MOV	REG+40,-(SP)	
	011006	013746	002536	MOV	REG+36,-(SP)	
	011012	013746	002534	MOV	REG+34,-(SP)	
	011016	013746	002532	MOV	REG+32,-(SP)	
	011022	012746	003264	MOV	#FRMT05,-(SP)	
	011026	012746	000010	MOV	#10,-(SP)	
	011032	010600		MOV	SP,R0	
	011034	104415		TRAP	C\$PNTX	
	011036	062706	000022	ADD	#22,SP	
45						
46	011042	005737	002416	TST	RH70	:IS IT RH70 CONTROLLER ?
47	011046	001424		BEQ	1\$:BR IF NO
48						:PRINT 'RPBAE RPCS3'
49	011050	012746	003346	MOV	#FRMT06,-(SP)	
	011054	012746	000001	MOV	#1,-(SP)	
	011060	010600		MOV	SP,R0	
	011062	104415		TRAP	C\$PNTX	
	011064	062706	000004	ADD	#4,SP	
50	011070	013746	002552	MOV	REG+52,-(SP)	
	011074	013746	002550	MOV	REG+50,-(SP)	
	011100	012746	003366	MOV	#FRMT07,-(SP)	
	011104	012746	000003	MOV	#3,-(SP)	
	011110	010600		MOV	SP,R0	

011112	104415		TRAP	C\$PNTX	
011114	062706	000010	ADD	#10,SP	
51 011120	000207		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
24
25
26
27
28
29

```
*****  
: 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 'USRBUF' (OUTPUT IS CONTROLLED BY 'NOROOM',  
: INDICATING THAT THE BSF IS ALREADY FULL)  
*  
: CALL  
: JSR PC,LODBSF ;CALL ROUTINE  
*****  
LODBSF: TST ENWTTD ;IS WRITE TD(S) ENABLED ?  
 BEQ 2$ ;BR IF NO  
 TSTB NOROOM ;ANY ROOM IN THE USR BSF ?  
 BMI 2$ ;BR IF NO  
 CMP R2,#ENDUSR ;IS USR BSF BUFFER FULL ?  
 BHI 1$ ;BR IF YES  
 MOV TMPBSF,(R2)+ ;LOG THE BSF DATA  
 MOV TMPBSF+2,(R2)+ ;CYLINDER/TRACK/SECTOR  
 BR 2$ ;AND FOR NOW, TAKE THE RETURN  
25 011156 112737 177777 002241 1$: MOVB #-1,NOROOM ;MARK NO ROOM IN BAD SECTOR FILE  
 ;PRINT '** WARNING ** USR BAD SECTOR FILE  
 ; BUFFER IS FULL NOW'  
28 011164 012746 006016 MOV #WARN,-(SP)  
 011170 012746 005237 MOV #FRMT63,-(SP)  
 011174 012746 000002 MOV #2,-(SP)  
 011200 010600 MOV SP,R0  
 011202 104417 TRAP C$PNTF  
 011204 062706 000006 ADD #6,SP  
29 011210 000207 2$: RTS PC ;RETURN TO THE 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
29
30
31
32
33
34
35
36
37
38
39

.SBTTL WRITE BUFFER POSTFIX MODULE

:THIS MODULE PERFORMS THE BUFFER POSTFIXING WHEN AN INTERLEAVED FORMAT
:OPERATION WAS PERFORMED. THE WRITE CHECK OPERATION IS PERFORMED IN A LINEAR
:FASHION, IE: SECTOR (N), SECTOR (N+1), SECTOR (N+2), SECTOR (N+...).
:*
:INPUTS ARE: INTLEV, LINBUF, WCKHD, WCKHDR, TKWCNT
:*
:OUTPUTS ARE: BADDR, FJUNCTN, ERRFNC, RPOF, NEGWRD
:*
:THIS ROUTINE IS CALLED FROM THE 'DRIVER'.
:*****

WRITCK: TSTB INTLEV ;INTERLEAVE MODE ?
BEQ 1\$;BR IF NO
JSR PC,CHABUF ;RE-SHUFFLE INTERLEAVED DATA AND
MOV #LINBUF,BADDR ;GET POINTER TO BEGINNING OF LINEAR BUFFER
1\$: MOV #WCKHD,FUNCTN ;SETUP A WRITE-CHECK HEADERS OPERATION
MOV #WCKHDR,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
BIS #CMOD,@RPOF ;SET COMMAND MODIFIER BIT
MOV TKWCNT,NEGWRD ;AND SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
RTS PC ;NOW TAKE THE RETURN

: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 B 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

011264
011264 010146
011266 010246
011270 010346
011272 010446
011274 010546
40 011276 012701 032250
41 011302 012702 034530
42 011306 012703 000002
43 011312 113705 002306
44 011316 005205
45 011320 006205
46 011322 012704 000006
47 011326 012122
48 011330 005304
49 011332 003375
50 011334 062701 000014
51 011340 005305
52 011342 003367

```
53 011344 012701 032264      MOV      #HDRBLK+12.,R1    :GET SECOND HALF OF BUFFER
54 011350 005303              DEC      R3                :SECOND PASS, R3=1
55 011352 003357              BGT     1$                 :IF > 0, IT TIME FOR SECOND PASS
56 011354 012605      MOV      (SP)+,R5        ::POP STACK INTO R5
    011356 012604      MOV      (SP)+,R4        ::POP STACK INTO R4
    011360 012603      MOV      (SP)+,R3        ::POP STACK INTO R3
    011362 012602      MOV      (SP)+,R2        ::POP STACK INTO R2
    011364 012601      MOV      (SP)+,R1        ::POP STACK INTO R1
57 011366 000207      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
24
25
26
27
28
29
30
31

```

:*****
:THIS MODULE COLLECTS THE USER INPUT TO MODIFY AN EXISTING TRACK DESCRIPTOR IT
:THEN TAKES THE USER VALUES AND MERGES THEM INTO AN EXISTING OR NEW TD RECORD.
:
:INPUTS ARE:  TDCPY3 - TDCPY6, DESCYL, DESTRK, #NULL
:
:OUTPUTS ARE: TDCPY3 - TDCPY6, TDWRD1 - TDWRD6, TEXT, ECCWRD, SECADD
:
:THIS MODULE IS CALLED BY: MAIN
:*****
    
```

```

INPUTD:
MOV    R0,-(SP)      ;;PUSH R0 ON STACK
MOV    R1,-(SP)      ;;PUSH R1 ON STACK
                ;;PRINT 'MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE)?'
TRAP   CS$GMAN
BR     10000$
.WORD  MODBY
.WORD  T$CODE
.WORD  MSG2
.WORD  T$LLOLIM
.WORD  T$HILIM
10000$:
                ;;PRINT 'CYLINDER ADDRESS (D) 0 ?'
TRAP   CS$GMAN
BR     10001$
.WORD  DESCYL
.WORD  T$CODE
.WORD  MSG38
.WORD  177777
.WORD  T$LLOLIM
.WORD  T$HILIM
10001$:
                ;;PRINT 'TRACK ADDRESS (D) 0 ?'
TRAP   CS$GMAN
BR     10002$
.WORD  DESTRK
.WORD  T$CODE
.WORD  MSG39
.WORD  177400
.WORD  T$LLOLIM
.WORD  T$HILIM
10002$:
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    #CURENT,TEXT ;;LOAD TEXT WITH ADDRESS OF MESSAGE
        JSR    PC,TDDUMP  ;;GO DUMP THE TRACK DESCRIPTOR
        CMP    MODBY,#1  ;;MODIFY BY WORDS, INDEX OR CHANGE MODE ?
        BLT   5$         ;;BR IF WORDS
        BEQ   6$         ;;BR IF INDEX
        BEQ   6$         ;;MUST BE CHANGE MODE
    
```

```

011370
011370 010046
011372 010146
011374 104443
011376 000406
011400 002176
011402 000052
011404 006334
011406 000003
011410 000000
011412 000002
011414
011414 104443
011416 000406
011420 002304
011422 000052
011424 012214
011426 177777
011430 000000
011432 001166
011434
011434 104443
011436 000406
011440 002244
011442 000052
011444 012235
011446 177400
011450 000000
011452 000037
011454
011454 004737 620466
011460 000401
011462 000402
011464 004737 025112
011470 012737 005640 002344 2$:
011476 004737 015014
011502 023727 002176 000001 3$:
011510 002507
011512 001542
    
```

```

32 011514 005037 030220      CLR      TDCPY3      ;PUT DUMMY NULL IN TD WORD #3
38 011520 005037 030222      CLR      TDCPY4      ;PUT DUMMY NULL IN TD WORD #4
   011524 005037 030224      CLR      TDCPY5      ;PUT DUMMY NULL IN TD WORD #5
   011530 005037 030226      CLR      TDCPY6      ;PUT DUMMY NULL IN TD WORD #6
42                                     ;PRINT 'TYPE <CR> TO INPUT (0=140000) DATA;'
43 011534 012746 004146      MOV      #FRMT24,-(SP)
   011540 012746 000001      MOV      #1,-(SP)
   011544 010600      MOV      SP,R0
   011546 104414      TRAP     C$PNTB
   011550 062706 000004      ADD      #4,SP
44                                     ;PRINT 'TD WORD#3 (D) 0 ?'
45 011554 104443      TRAP     C$GMAN
   011556 000406      BR       10003$
   011560 030220      .WORD   TDCPY3
   011562 000052      .WORD   T$CODE
   011564 012332      .WORD   MSG43
   011566 177777      .WORD   177777
   011570 000166      .WORD   T$LOLIM
   011572 042077      .WORD   T$HILIM
   011574                                     10003$:
46 011574 005737 030220      TST      TDCPY3      ;NULL ENTRY ?
47 011600 001436      BEQ      4$          ;BR IF YES
48                                     ;PRINT 'TD WORD#4 (D) 0 ?'
49 011602 104443      TRAP     C$GMAN
   011604 000406      BR       10004$
   011606 030222      .WORD   TDCPY4
   011610 000052      .WORD   T$CODE
   011612 012344      .WORD   MSG44
   011614 177777      .WORD   177777
   011616 000001      .WORD   T$LOLIM
   011620 042077      .WORD   T$HILIM
   011622                                     10004$:
50 011622 005737 030222      TST      TDCPY4      ;NULL ENTRY ?
51 011626 001423      BEQ      4$          ;BR IF YES
52                                     ;PRINT 'TD WORD#5 (D) 0 ?'
53 011630 104443      TRAP     C$GMAN
   011632 000406      BR       10005$
   011634 030224      .WORD   TDCPY5
   011636 000052      .WORD   T$CODE
   011640 012356      .WORD   MSG45
   011642 177777      .WORD   177777
   011644 000001      .WORD   T$LOLIM
   011646 042077      .WORD   T$HILIM
   011650                                     10005$:
54 011650 005737 030224      TST      TDCPY5      ;NULL ENTRY ?
55 011654 001410      BEQ      4$          ;BR IF YES
56                                     ;PRINT 'TD WORD#6 (D) 0 ?'
57 011656 104443      TRAP     C$GMAN
   011660 000406      BR       10006$
   011662 030226      .WORD   TDCPY6
   011664 000052      .WORD   T$CODE
   011666 012370      .WORD   MSG46
   011670 177777      .WORD   177777
   011672 000001      .WORD   T$LOLIM
   011674 042077      .WORD   T$HILIM
   011676                                     10006$:
58 011676 013737 030220 030204 4$: MOV      TDCPY3,TDWRD3 ;GET TD WORD #3
    
```


WRITE BUFFER POSTFIX MODULE

59	011704	013737	030222	030206	MOV	TDCPY4,TDWRD4	:GET TD WORD #4
60	011712	013737	030224	030210	MOV	TDCPY5,TDWRD5	:GET TD WORD #5
61	011720	013737	030226	030212	MOV	TDCPY6,TDWRD6	:GET TD WORD #6
62	011726	000451			BR	8\$	
63	011730				5\$:		:PRINT 'SECTOR ADDRESS (D) ?'
64	011730	104443			TRAP	C\$GMAN	
	011732	000406			BR	10007\$	
	011734	002366			.WORD	SECADD	
	011736	000042			.WORD	T\$CODE	
	011740	012313			.WORD	MESG42	
	011742	177777			.WORD	177777	
	011744	000000			.WORD	T\$LOLIM	
	011746	000061			.WORD	T\$HILIM	
	011750				10007\$:		
65	011750	104443					:PRINT 'WORD IN SECTOR (D) ?'
66	011752	000406			TRAP	C\$GMAN	
	011754	002364			BR	10010\$	
	011756	000042			.WORD	ECCWRD	
	011760	012253			.WORD	T\$CODE	
	011762	177777			.WORD	MESG40	
	011764	000000			.WORD	177777	
	011766	000401			.WORD	T\$LOLIM	
	011770				.WORD	T\$HILIM	
					10010\$:		
67	011770	004737	014372		JSR	PC,RELTIME	:CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
68	011774	023737	002364	002340	CMP	ECCWRD,TDLMTS+2	:WILL BEGINNING OF DEFECT BE IN GAP 2 ?
69	012002	003016			BGT	7\$:BR IF NO
70	012004	013737	002340	002364	MOV	TDLMTS+2,ECCWRD	:GET UPPER LIMIT FOR HDR 0 MOVED AND
71	012012	005237	002364		INC	ECCWRD	:ADD ONE TO IT.
72	012016	000410			BR	7\$:AND GO INSERT IT INTO TD.
73	012020				6\$:		:PRINT 'WORDS FROM INDEX (D) ?'
74	012020	104443			TRAP	C\$GMAN	
	012022	000406			BR	10011\$	
	012024	002364			.WORD	ECCWRD	
	012026	000042			.WORD	T\$CODE	
	012030	012272			.WORD	MESG41	
	012032	177777			.WORD	177777	
	012034	000166			.WORD	T\$LOLIM	
	012036	042553			.WORD	T\$HILIM	
	012040				10011\$:		
75	012040	004737	014522		7\$:	JSR	PC,INSERT
76	012044	005737	030244		TST	TMPBSF	:DO WE HAVE ROOM IN THE TD RECORD ?
77	012050	100045			BPL	14\$:BR IF NO
78							
79	012052	012700	030204		8\$:	MOV	#TDWRD3,R0
80	012056	012701	000004		MOV	#4,R1	:GET POINTER TO BEGINNING OF TD BUFFER
81	012062	005720			9\$:	TST	(R0)+
82	012064	001003			BNE	10\$:IS THIS A DUMMY (0) ENTRY ?
83	012066	012760	140000	177776	MOV	#NULL,-2(R0)	:MAKE DUMMY (0) ENTRY A NULL (140000) ENTRY
84	012074	005301			10\$:	DEC	R1
85	012076	003371			BGT	9\$:DONE ALL WORDS YET ?
86							:BR IF NO
87	012100	013700	002222		MOV	ERRMAX,R0	:SETUP RETRY COUNT
88	012104	004737	025016		JSR	PC,SAVEID	:SAVE TD WRITE BUFFER JUST INCASE THERE
89							:IS RETRY ATTEMPT.
90	012110	004737	025054		11\$:	JSR	PC,RESTID
91	012114	004737	020222		JSR	PC,WRITID	:NOW RESTORE THE TD BUFFER
							:CALL THE WRITE TD SUBROUTINE

```

92 012120 000421 BR 14$ ;RETURN HERE IF EXCEEDED RETRY LIMIT
93 ;ELSE RETURN HERE, NO ERROR
94 012122 012737 177777 002242 MOV #-1,SUPRSS ;SUPPRESS THE ERROR MESSAGE OUTPUT
95 012130 005300 12$: DEC RO ;DID WE EXCEED RETRY ?
96 012132 002414 BLT 14$ ;BR IF YES
97 012134 003002 BGT 13$ ;BR IF NO
98 012136 005037 002242 CLR SUPRSS ;ALLOW ERROR MESSAGE TO BE OUTPUT
99 012142 004737 020466 13$: JSR PC,READTD ;CALL THE READ TD SUBROUTINE
100 012146 000770 BR 12$ ;RETURN HERE IF EXCEEDED RETRY LIMIT
101 ;ELSE RETURN HERE, NO ERROR
102 012150 012737 006000 002344 MOV #CHANGE,TEXT ;LOAD TEXT WITH ADDRESS OF MESSAGE
103 012156 004737 015014 JSR PC,TDDUMP ;GO DUMP THE TRACK DESCRIPTOR
104 012162 000401 BR 15$
105 012164 005725 14$: TST (R5)+ ;ADJUST R5 TO SKIP FORMAT TRACK, ON RETURN
106 012166 005037 002242 15$: CLR SUPRSS ;ALLOW ERROR MESSAGE TO BE OUTPUT
107 012172 012601 MOV (SP)+,R1 ;POP STACK INTO R1
108 012174 012600 MOV (SP)+,R0 ;POP STACK INTO R0
109 ;PRINT 'DO YOU WANT TO MODIFY ANYMORE TD'S (L) N ?'
109 012176 104443 TRAP C$GMAN
109 012200 000404 BR 10012$
109 012202 002402 .WORD MORETD
109 012204 000130 .WORD T$CODE
109 012206 012402 .WORD MESG47
109 012210 000001 .WORD 1
110 012212 000207 10012$: RTS PC ;TAKE THE RETURN
111
115 012214 103 131 114 MESG38: .ASCIZ /CYLINDER ADDRESS/
116 012235 124 122 101 MESG39: .ASCIZ /TRACK ADDRESS/
117 012253 127 117 122 MESG40: .ASCIZ /WORD IN SECTOR/
118 012272 127 117 122 MESG41: .ASCIZ /WORDS FROM INDEX/
119 012313 123 105 103 MESG42: .ASCIZ /SECTOR ADDRESS/
120 012332 124 104 040 MESG43: .ASCIZ /TD WORD#3/
121 012344 124 104 040 MESG44: .ASCIZ /TD WORD#4/
122 012356 124 104 040 MESG45: .ASCIZ /TD WORD#5/
123 012370 124 104 040 MESG46: .ASCIZ /TD WORD#6/
124 012402 104 117 040 MESG47: .ASCIZ /DO YOU WANT TO MODIFY ANYMORE TD'S/
125
126 .EVEN
    
```



```

1      .SBTTL PROGRAM UTILITIES
2      .SBTTL RETRY COUNT MODULE
3
4      :*****
5      :THIS ROUTINE MAINTAINS THE RETRY COUNTER. RETRY MAXIMUM IS 5. THAT IS, 5
6      :ITERATIONS MUST OCCUR BEFORE AN ERROR IS REPORTED. WHEN AN ERROR IS REPORTED,
7      :THE DRIVE MUST BE FIXED, THE PROGRAM CAN DO NO MORE!!
8
9      :INPUTS ARE: ERRMSK
10
11     :OUTPUTS ARE: ERRMSK, ERRDNT, ERRITL
12
13     :CALL
14     :      JSR      PC,RETRY      :CALL THE ROUTINE
15     :      -----      :RETURN HERE IF EXCEEDED MAX RETRY LIMIT
16     :      -----      :ELSE RETURN HERE, IF RTYCNT >= 1 THEN ERROR
17     :                          IF RTYCNT = 0 THEN NO ERROR
18
19     :THIS MODULE IS MAINLY CALLED BY: MAKTRK, DRVINI, RDBSF, WRTBSF, WRITTD,
20     :      READTD, HDSCAN, DASCAN
21     :*****
22
23     012446 062716 000002 RETRY: ADD #2,(SP) :ADJUST FOR GOOD RETURN
24     012452 004737 017012 JSR PC,SAVRPR :GET THE REGISTER SNAPSHOT NOW
25     012456 005737 002264 TST ERRMSK :DID WE GET ANY ERRORS ?
26     012462 001414 BEQ 1$ :BR IF NO
27     012464 005237 002266 INC RTYCNT :INCREMENT RETRY COUNTER AND
28     012470 005237 002262 INC ERRITL :THE TOTAL ERROR COUNTER
29     012474 023737 002266 002222 CMP RTYCNT,ERRMAX :DID WE EXCEED THE MAX RETRY LIMIT ON ERROR ?
30     012502 103411 BLO 2$ :BR IF NO
31     012504 162716 000002 SUB #2,(SP) :ADJUST FOR EXCEEDED RETRY LIMIT RETURN
32     012510 004737 012752 JSR PC,ERRORS :REPORT THE ERROR NOW!
33
34     012514 005037 002266 1$: CLR RTYCNT :RESET RETRY COUNT TO 0
35     012520 005737 002264 TST ERRMSK :ANY MORE ERRORS?
36     012524 001404 BEQ 3$ :BR IF NO
37     012526 005037 002264 2$: CLR ERRMSK :RESET NO ERROR STATUS
38     012532 004737 012540 JSR PC,RESET :INITIALIZE THE MASSBUS
39     012536 000207 3$: RTS PC :EXIT
40
41     :*****
42     :THIS MODULE, WHEN CALLED DOES A CLEAR TO THE DRIVE IT IS DEPENDENT ON THE
43     :MODULE "SEIZE" TO RELOAD THE DRIVE NUMBER AFTER THE RESET.
44
45     :INPUTS ARE: NONE
46
47     :OUTPUTS ARE: RPCS1, RPOF, RPMR1
48
49     :THIS MODULE IS CALLED BY: RETRY
50     :*****
51
52     012540 004737 012602 RESET: JSR PC,SEIZE :NOW GET THE DRIVE
53     012544 012777 000021 167652 MOV #READIN,@RPCS1 :DO A READ IN PRESET
54     012552 012777 010000 167676 MOV #FMT16,@RPOF :SET UP FOR 16 BIT WORD
55     012560 105737 002234 TSTB CEMODE :ARE WE IN MAINTENANCE MODE?
56     012564 001403 BEQ 1$ :NOT IF ZERO
57     012566 052777 100000 167654 BIS #BIT15,@RPMR1 :YES, RESET THE MAINTENANCE BIT

```

```

RETRY COUNT MODULE

58 012574 004737 010156      1$:   JSR   PC,READY      ;WAIT FOR DRIVE READY NOW!
59 012600 000207              RTS   PC                ;NOW RETURN TO MAIN
60
61
62
63
64
65
66
67
68
69
70
71
72
73 012602 052777 000040 167624 SEIZE:  BIS   #CLR,@RPCS2    ;CLEAR THE MASSBUS
74 012610 013777 002420 167616      MOV   DRVNO,@RPCS2    ;RELOAD THE DRIVE NUMBER
75 012616 105777 167614      TSTB  @RPDS           ;IS THIS DRIVE SEIZED BY ANOTHER PORT?
76 012622 100435              BMI   3$              ;NO, JUST TAKE RETURN
77 012624 005077 167606      CLR   @RPDS           ;ISSUE A DRIVE REQUEST
78 012630 010246              MOV   R2,-(SP)        ;SAVE R2
79 012632 012702 000012      MOV   #10.,R2         ;LOAD R2 WITH AN OVERAL ITERATION COUNT
80 012636 005777 167574      1$:   TST   @RPDS       ;NOW WAIT FOR THE OTHER PORT TO RELEASE
81 012642 001021              BNE   2$              ;NOT YET IF RPDS = 0
82 012644 004737 012720      JSR   PC,WAIT         ;AND REDUCE THE ITERATION COUNT
83 012650 005302              DEC   R2              ;LOOK AGAIN FOR DRIVE PRESENT
84 012652 001371              BNE   1$              ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
85 012654 012737 000003 002264      MOV   #3,ERRMSK      ;PUSH SUPRSS ON STACK
86 012662 013746 002242      MOV   SUPRSS,-(SP)   ;ALLOW ERROR MESSAGE TO BE OUTPUT
87 012666 005037 002242      CLR   SUPRSS         ;GET THE REGISTER SNAPSHOT NOW
88 012672 004737 017012      JSR   PC,SAVRPR      ;AND REPORT THE ERROR
89 012676 004737 012752      JSR   PC,ERRORS      ;POP STACK INTO SUPRSS
90 012702 012637 002242      MOV   (SP)+,SUPRSS
91
92 012706 112777 000377 167526 2$:   MOVB  #377,@RPAS     ;CLEAR ANY UNWANTED ATTENTION BITS
93 012714 012602              MOV   (SP)+,R2       ;RESTORE R2
94 012716 000207              3$:   RTS   PC          ;AND TAKE RETURN
95
96
97
98
99
100
101
102
103
104 012720
105 012720 012727 000372      WAIT:  MOV   #250.,(PC)+
106 012724 000000              .WORD 0
107 012726 013727 002116      MOV   L$DLY,(PC)+
108 012732 000000              .WORD 0
109 012734 005367 177772      DEC   -6(PC)
110 012740 001375              BNE   -4
111 012742 005367 177756      DEC   -22(PC)
112 012746 001367              BNE   -20
113 012750 000207              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
:*****

:*****
:THIS MODULE IS USED TO STALL IN LOOPS WHICH MUST WAIT FOR THE RPO7
:MICROPROCESSORS. THE WAIT IS A FIXED TIME PERIOD, AND CANNOT BE CHANGED
:DYNAMICALLY. THERE ARE NO SIDE EFFECTS.
:CALL
:
:MODULE IS CALLED BY: SEIZE
:*****

```



```

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 012752 005737 002264  ERRORS: TST      ERRMSK      ;DID WE GET ANY ERRORS?
15 012756 001430          BEQ      5$          ;IF ERRMSK = 0, NO!
16 012760 005737 002242  TST      SUPRSS      ;ALLOW ERROR MESSAGES ?
17 012764 100425          BMI      5$          ;IF MINUS, NO....
18 012766 010146          MOV      R1,-(SP)     ;:PUSH R1 ON STACK
19 012770 010246          MOV      R2,-(SP)     ;:PUSH R2 ON STACK
20 012772 012701 013042  MOV      #6$,R1      ;GET THE TOP OF THE ERROR DISPATCH FILE
21 012776 012702 000001  MOV      #1,R2       ;AND SET UP THE ERROR NUMBER MASK.
22 013002 023702 002264  1$:  CMP      ERRMSK,R2   ;MATCH?
23 013006 001407          BEQ      2$          ;TAKE BRANCH OF DS - REPORT FIND NOW!
24 013010 095202          INC      R2          ;GET NEXT ERROR POSITION
25 013012 062701 000012  ADD      #12,R1      ;MOVE THE DISPATCH POINTER TO NEXT MESSAGE
26 013016 020127 013302  CMP      R1,#7$     ;ARE WE AT THE END OF ERROR TABLE ?
27 013022 103767          BLO      1$          ;BR IF NO
28 013024 000403          BR       4$
29 013026 000111          2$:  JMP      (R1)      ;REPORT FIND NOW
30
31 013030 005237 002260  3$:  INC      ERTTL     ;ADD ONE TO THE ERROR TOTAL
32 013034          4$:  MOV      (SP)+,R2   ;:POP STACK INTO R2
33 013034 012602          MOV      (SP)+,R1   ;:POP STACK INTO R1
34 013036 012601          5$:  RTS      PC      ;RETURN
35 013040 000207
36
37
38 013042          6$:  ERRHRD  16,EM16,ERR3 ;TD RELOCATION ERROR ;ERRMSK=1
39 013042 104456          TRAP    C$ERHRD
40 013044 000020          .WORD  16
41 013046 007141          .WORD  EM16
42 013050 007704          .WORD  ERR3
43 013052 000766          BR      3$          ;RETURN
44
45 013054          ERRHRD  3,EM3,ERRO ;COMPOSITE ERROR ;ERRMSK=2
46 013054 104456          TRAP    C$ERHRD
47 013056 000003          .WORD  3
48 013060 006600          .WORD  EM3
49 013062 007406          .WORD  ERRO
50 013064 000761          BR      3$          ;RETURN
51
52 013066          ERRHRD  4,EM4,ERRO ;DRIVE HUNG ;ERRMSK=3
53 013066 104456          TRAP    C$ERHRD
54 013070 000004          .WORD  4
55 013072 006624          .WORD  EM4
56 013074 007406          .WORD  ERRO
57 013076 000754          BR      3$          ;RETURN
    
```

46					
47	013100		ERRHRD	5,EM5,ERRO	:DRIVE WRITE LOCKED ;ERRMSK=4
	013100	104456	TRAP	C\$ERHRD	
	013102	000005	.WORD	5	
	013104	006664	.WORD	EM5	
	013106	007406	.WORD	ERRO	
48	013110	000747	BR	3\$:RETURN
49					
50	013112		ERRHRD	6,EM6,ERRO	:DRIVE OFFLINE ;ERRMSK=5
	013112	104456	TRAP	C\$ERHRD	
	013114	000006	.WORD	6	
	013116	006707	.WORD	EM6	
	013120	007406	.WORD	ERRO	
51	013122	000742	BR	3\$:RETURN
52					
53	013124		ERRHRD	7,EM7,ERRO	:UNEXPECTED ATTN ;ERRMSK=6
	013124	104456	TRAP	C\$ERHRD	
	013126	000007	.WORD	7	
	013130	006725	.WORD	EM7	
	013132	007406	.WORD	ERRO	
54	013134	000735	BR	3\$:RETURN
55					
56	013136		ERRHRD	10,EM10,ERRO	:WRITE CHECK ERROR ;ERRMSK=7
	013136	104456	TRAP	C\$ERHRD	
	013140	000012	.WORD	10	
	013142	006756	.WORD	EM10	
	013144	007406	.WORD	ERRO	
57	013146	000730	BR	3\$:RETURN
58					
59	013150		ERRHRD	11,EM11,ERR5	:FORMAT VERIFY ERROR ;ERRMSK=10
	013150	104456	TRAP	C\$ERHRD	
	013152	000013	.WORD	11	
	013154	007004	.WORD	EM11	
	013156	007732	.WORD	ERR5	
60	013160	000723	BR	3\$:RETURN
61					
62	013162		ERRHRD	1,EM1,ERR1	:TD ADDRESS INCORRECT ;ERRMSK=11
	013162	104456	TRAP	C\$ERHRD	
	013164	000001	.WORD	1	
	013166	006527	.WORD	EM1	
	013170	007440	.WORD	ERR1	
63	013172	000716	BR	3\$:RETURN
64					
65	013174		ERRHRD	2,EM2,ERR2	:TD FORMAT INCORRECT ;ERRMSK=12
	013174	104456	TRAP	C\$ERHRD	
	013176	000002	.WORD	2	
	013200	006554	.WORD	EM2	
	013202	007562	.WORD	ERR2	
66	013204	000711	BR	3\$:RETURN
67					
68	013206		ERRHRD	12,EM12,ERRO	:DRY DIDN'T RESET ;ERRMSK=13
	013206	104456	TRAP	C\$ERHRD	
	013210	000014	.WORD	12	
	013212	007030	.WORD	EM12	
	013214	007406	.WORD	ERRO	
69	013216	000704	BR	3\$:RETURN
70					


```
71 013220          ERRHRD 13,EM13,ERRO ;TRANSFER ERROR      ;ERRMSK=14
    013220 104456  TRAP    C$ERHRD
    013222 000015  .WORD 13
    013224 007067  .WORD EM13
    013226 007406  .WORD ERRO
72 013230 000677  BR      3$          ;RETURN
73
74 013232          ERRHRD 14,EM14,ERRO ;SPECIAL CONDITION   ;ERRMSK=15
    013232 104456  TRAP    C$ERHRD
    013234 000016  .WORD 14
    013236 007112  .WORD EM14
    013240 007406  .WORD ERRO
75 013242 000672  BR      3$          ;RETURN
76
77 013244          ERRHRD 15,EM15,ERRO ;NOT USED             ;ERRMSK=16
    013244 104456  TRAP    C$ERHRD
    013246 000017  .WORD 15
    013250 007140  .WORD EM15
    013252 007406  .WORD ERRO
78 013254 000665  BR      3$          ;RETURN
79
80 013256          ERRHRD 17,EM17,ERRO ;UNRECOVERABLE ERROR ;ERRMSK=17
    013256 104456  TRAP    C$ERHRD
    013260 000021  .WORD 17
    013262 007216  .WORD EM17
    013264 007406  .WORD ERRO
81 013266 000660  BR      3$          ;RETURN
82
83 013270          ERRHRD 20,EM20,ERRO ;RANDOM WCE'S DURING  ;ERRMSK=20
    013270 104456  TRAP    C$ERHRD
    013272 000024  .WORD 20
    013274 007263  .WORD EM20
    013276 007406  .WORD ERRO
84 013300 000653  BR      3$          ;RETURN
88
89 013302          7$:
```

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

```

:*****
:THIS MODULE IS USED ANYTIME A FORMAT TRACK OR VERIFY TRACK COMMAND IS ISSUED.
:THE MODULE CREATES THE BUFFER USED FOR THE OPERATION AND DOES THE SETUP FOR
:THE COMMAND EXECUTION. IT ALSO EXECUTES THE COMMAND AND ALLOWS A RETRY IF
:THE COMMAND FAILS.
:
:INPUTS ARE:  FUNCTN, TKWCNT
:
:OUTPUTS ARE:  ERRFNC, BADDR, NEGWRD, RPOF
:
:CALL
:      JSR      PC,MAKTRK      ;NOW EXECUTE THE COMMAND
:      -----
:      -----
:      -----
:      ;RETURN HERE IF EXCEEDED RETRY LIMIT
:      ;ELSE RETURN HERE
:
:THIS MODULE IS CALLED BY: MAIN
:*****
    
```

```

20 013302
013302 010046
013304 010146
013306 010246
013310 010346
013312 010446
013314 010546
21 013316 004737 015416
22 013322 004737 016610
23 013326 012605
013330 012604
013332 012603
013334 012602
013336 012601
013340 012600
24 013342 022737 000063 002370 1$:
25 013350 001004
26 013352 012737 005510 002254
27 013360 000413
28
29 013362 022737 000073 002370 2$:
30 013370 001007
31 013372 012737 005523 002254
32 013400 012737 033400 002230
33 013406 000403
34
35 013410 012737 032250 002230 3$:
36 013416 013737 002350 002362 4$:
37 013424 052777 100000 167024
38 013432 004737 010166
39 013436 004737 012446
40 013442 000405
41
42 013444 005737 002266
43 013450 001334
44 013452 062716 000002
45 013456 000207
    
```

```

MAKTRK:
MOV R0,-(SP) ;:PUSH R0 ON STACK
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
JSR PC,CALCTD ;:DO THE DEFECT LOCATING NOW
JSR PC,MERGE ;:NOW PUT THE DEFECTS IN THE CORRECT SECTOR
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
MOV (SP)+,R0 ;:POP STACK INTO R0
CMP #FMTRK,FUNCTN ;:IS THIS A FORMAT COMMAND?
BNE 2$ ;:BR IF NO
MOV #FORMAT,ERRFNC ;:SAVE FUNCTION FOR ERROR ROUTINE
BR 3$ ;:AND SKIP NEXT
2$:
CMP #RDHD,FUNCTN ;:IS THIS A VERIFY FUNCTION?
BNE 3$ ;:BR IF NO (MUST BE A WRITE CHECK)
MOV #RHDATA,ERRFNC ;:SAVE FUNCTION FOR ERROR ROUTINE
MOV #TDSBLK,BADDR ;:GET POINTER TO BEGINNING OF TD BLOCK
BR 4$ ;:AND GO-ON
3$:
MOV #HDRBLK,BADDR ;:GET POINTER TO BEGINNING OF HEADER BLOCK
MOV TKWCNT,NEGWRD ;:SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
BIS #CMOD,@RPOF ;:SET COMMAND MODIFIER BIT
JSR PC,DRIVER ;:EXECUTE THE OPERATION NOW
JSR PC,RETRY ;:SEE IF WE HAD ANY ERRORS
BR 5$ ;:RETURN HERE IF EXCEEDED RETRY LIMIT
;:ELSE RETURN HERE
TST R1YCNT ;:DID WE HAVE AN ERROR ?
BNE 1$ ;:BR IF YES
ADD #2,(SP) ;:ADJUST FOR GOOD RETURN
RTS PC ;:TAKE RETURN NOW
5$:
    
```



```

1
2
3
4
5
6
7
8
9
10 013460 004737 012602          DRVINI: JSR      PC,SEIZE          ;SEIZE THE PORT
11                                     ;PRINT 'DRIVE X'
12 013464 013746 002420          MOV      DRVNO, -(SP)
    013470 012746 003405          MOV      #FRMT10, -(SP)
    013474 012746 000002          MOV      #2, -(SP)
    013500 010600                    MOV      SP, R0
    013502 104417                    TRAP     C$PNTF
    013504 062706 000006          ADD      #6, SP
13 013510 005777 166722          TST     @RPDS                    ;DO A READ OF THE DRIVE STATUS REG
14 013514 032777 010000 166712  BIT     #NED, @RPCS2              ;DO WE HAVE NED?
15 013522 001405                    BEQ     1$                        ;IF ZERO, DRIVE DOES EXIST
16                                     ;REPORT 'NON-EXISTENT DRIVE'
17 013524 104455                    TRAP     C$ERDF
    013526 000001                    .WORD   1
    013530 007341                    .WORD   EM21
    013532 000000                    .WORD   0
18 013534 000572                    BR      NORUN                    ;TAKE THE BAD BRANCH, THE DRIVE IS NO GOOD!
19
20 013536 032777 000004 166672  1$:  BIT     #ILEV, @RPDS              ;IS THIS DRIVE INTERLEAVE ENABLED?
21 013544 001414                    BEQ     2$                        ;IF BIT = 0, NO
22 013546 112737 000377 002237  MOVB    #377, INTLEV              ;SET THE INTERLEAVE ENABLED MASK
23                                     ;PRINT ', INTERLEAVED'
24 013554 012746 004244          MOV      #FRMT31, -(SP)
    013560 012746 000001          MOV      #1, -(SP)
    013564 010600                    MOV      SP, R0
    013566 104417                    TRAP     C$PNTF
    013570 062706 000004          ADD      #4, SP
25 013574 000410                    BR      3$
26 013576                    2$:  ;PRINT ', NOT INTERLEAVED'
27 013576 012746 004216          MOV      #FRMT30, -(SP)
    013602 012746 000001          MOV      #1, -(SP)
    013606 010600                    MOV      SP, R0
    013610 104417                    TRAP     C$PNTF
    013612 062706 000004          ADD      #4, SP
28 013616 017746 166630          3$:  MOV      @RPDT, -(SP)              ;GET THE DRIVE TYPE REGISTER
29 013622 042716 004000          BIC     #BIT11, (SP)              ;CLEAR THE DRIVE REQUEST REQUIRED BIT
30 013626 022726 020042          CMP     #20042, (SP)+              ;IS THE DRIVE AN RP07?
31 013632 001405                    BEQ     4$                        ;BR IF YES
32                                     ;REPORT 'DRIVE NOT AN RP07'
33 013634 104455                    TRAP     C$ERDF
    013636 000002                    .WORD   2
    013640 007364                    .WORD   EM22
    013642 000000                    .WORD   0
34 013644 000526                    BR      NORUN
35
36 013646 012701 000004          4$:  MOV      #4, R1                    ;GET AN OVERALL ITERATION COUNT
37 013652 105777 166560          5$:  TSTB   @RPDS                    ;IS DRIVE READY SET ?
38 013656 100415                    BMI     6$                        ;BR IF YES
    
```

```

39 013660 004737 012720      JSR    PC,WAIT      ;WAIT A LITTLE LONGER
40 013664 005301              DEC    R1            ;DONE TRYING YET ?
41 013666 003371              BGT    5$           ;BR IF NO
42 013670 012737 000003 002264  MOV    #3,ERRMSK    ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
43 013676 012737 005567 002254  MOV    #DRVDRY,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
44 013704 004737 012752      JSR    PC,ERRORS    ;NOW REPORT THE ERROR
45 013710 000504              BR     NORUN        ;DRIVE IS NO GOOD
46
47 013712 032777 010000 166516 6$:  BIT    #MOL,@RPDS   ;IS THE DRIVE ONLINE?
48 013720 001014              BNE    7$           ;BR IF YES
49 013722 012737 005537 002254  MOV    #MOLINE,ERRFNC ;MARK THE CURRENT FUNCTION
50 013730 012737 000005 002264  MOV    #5,ERRMSK    ;LOG 'DRIVE OFFLINE ERROR' STATUS
51 013736 004737 012446      JSR    PC,RETRY     ;SEE IF WE HAD ANY ERRORS
52 013742 000471              BR     NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
53
54 013744 005737 002266      TST    RTYCNT       ;ELSE RETURN HERE
55 013750 001360              BNE    6$           ;DID WE HAVE AN ERROR ?
56
57 013752 005737 002166      7$:  TST    VRIFY       ;IS THIS VERIFY ONLY MODE ?
58 013756 001023              BNE    9$           ;BR IF YES
59 013760 005737 002200      TST    LIST        ;IS THIS LIST MODE ?
60 013764 001020              BNE    9$           ;BR IF YES
61 013766 032777 004000 166442 8$:  BIT    #WRL,@RPDS   ;IS THE DRIVE WRITE LOCKED ?
62 013774 001414              BEQ    9$           ;BR IF NO
63 013776 012737 005553 002254  MOV    #WLOCK,ERRFNC ;MARK THE WRITE LOCK TEST
64 014004 012737 000004 002264  MOV    #4,ERRMSK    ;LOG 'DRIVE WRITE LOCKED ERROR' STATUS
65 014012 004737 012446      JSR    PC,RETRY     ;SEE IF WE HAD ANY ERRORS
66 014016 000443              BR     NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
67
68 014020 005737 002266      TST    RTYCNT       ;ELSE RETURN HERE
69 014024 001360              BNE    8$           ;DID WE HAVE AN ERROR ?
70
71 014026 012737 005502 002254 9$:  MOV    #RECALI,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
72 014034 012777 000007 166362  MOV    #RECAL,@RPCS1 ;DO RECALIBRATE COMMAND
73 014042 105777 166370      TSTB   @RPDS        ;DID DRIVE READY RESET DURING RECAL ?
74 014046 100011              BPL    10$          ;BR IF YES
75 014050 012737 000013 002264  MOV    #13,ERRMSK   ;LOG 'DRIVE READY DIDN'T RESET ERROR' STATUS
76 014056 004737 012446      JSR    PC,RETRY     ;SEE IF WE HAD ANY ERRORS
77 014062 000421              BR     NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
78
79 014064 005737 002266      TST    RTYCNT       ;ELSE RETURN HERE
80 014070 001356              BNE    9$           ;DID WE HAVE AN ERROR ?
81
82 014072 012737 000007 002370 10$: MOV    #RECAL,FUNCTN ;SET UP FOR THE DRIVER
83 014100 004737 010166      JSR    PC,DRIVER    ;AND DO THE RECAL AGAIN
84 014104 004737 012446      JSR    PC,RETRY     ;SEE IF WE HAD ANY ERRORS
85 014110 000406              BR     NOGOOD       ;RETURN HERE IF EXCEEDED RETRY LIMIT
86
87 014112 005737 002266      TST    RTYCNT       ;ELSE RETURN HERE
88 014116 001365              BNE    10$          ;DID WE HAVE AN ERROR ?
89 014120 000405              BR     BUFINI       ;BR IF YES
90
91 014122 005237 002262      NORUN: INC    ERRTTL  ;ADD ONE TO THE ERROR TOTAL
92 014126 012737 177777 002256  NOGOOD: MOV    #-1,SOF SW ;LOAD THE FAILED STATUS
93 014134 012701 030244      BUFINI: MOV    #BEGBUF,R1 ;GET POINTER TO BEGINNING OF BUFFER SECTION
94 014140 005021              1$:  CLR    (R1)+       ;INITIALIZE ALL BUFFERS
95 014142 020127 117656      CMP    R1,#ENDBUF  ;DONE YET ?

```


96 014146 101774
97 014150 000207

BLOS 1\$
RTS PC

;BR IF NO

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 014152
014152 010246
014154 010346
014156 010546
18 014160 012702 000024
19 014164 012703 002424
20 014170
21 014170 013700 002404
014174 104442
014176 010005
22 014200 103034
23 014202 011346
24 014204 011546
25 014206 166616 000002
26 014212 061623
27 014214 005302
28 014216 001375
29 014220 004737 010014
30 014224 005737 002416
31 014230 001406
32 014232 013702 002414
33 014236 061502
34 014240 010223
35 014242 005722
36 014244 010213
37
38 014246 022626
39 014250 012537 002406
40 014254 012537 002410
41 014260 012537 002412
42 014264 011537 002420
43 014270 000406
44
45 014272 005237 002404
46 014276 023737 002404 002012
47 014304 002731
48 014306
014306 012605
014310 012603
014312 012602
49 014314 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: NONE
:
:OUTPUTS ARE: RPADD, RHTYPE, RPBAE, RPCS3, DRIVE, RPCS1
:*
:THIS MODULE IS CALLED BY: THE INITIALIZATION CODE
:*****
    
```

```

TABLED::
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$:
MOV UNIT,R0
TRAP C$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$:
CMP (SP)+,(SP)+  :DONE, RESTORE THE STACK
MOV (R5)+,RPADR  :SAVE RPCS1 BASE ADDRESS
MOV (R5)+,RPVEC  :SAVE INTERRUPT VECTOR ADDRESS
MOV (R5)+,RPVEC+2 :SAVE INTERRUPT PRIORITY
MOV (R5),DRVNO   :SETUP DRIVE NUMBER FOR UNIT N
BR 5$            :SKIP NEXT
4$:
INC UNIT
CMP UNIT,L$UNIT  :TRY THE NEXT UNIT
BLT 1$          :IS THIS THE LAST UNIT TO TRY ?
5$:
MOV (SP)+,R5     ::PCP STACK INTO R5
MOV (SP)+,R3     ::POP STACK INTO R3
MOV (SP)+,R2     ::POP STACK INTO R2
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

```
*****  
: THIS MODULE IS USED IN THE SCANNING PROCESS TO PROVIDE ADDRESSES TO THE EXEC  
: WHEN A SECOND OR CONSECUTIVE ITERATION IS NEEDED DURING A SCAN OPERATION.  
: THE ADDRESSES ARE LOCATED IN THE BUFFER "DBUFF" AND ARE PLACED THERE AS A  
: RESULT OF A PREVIOUS SCAN OPERATION.  
:  
: INPUTS ARE:   TBLPTR  
:  
: OUTPUTS ARE:  DESCYL, DESTRK  
:  
: THIS MODULE IS CALLED BY: MAIN  
:*****
```

```
15 014316 023737 002252 002232 GETNEX: CMP      TBLPTR,LASLOC  :DONE ?  
16 014324 103021          BHIS      1$          :IF =>, YES!!  
17 014326 017737 165720 002304      MOV      @TBLPTR,DESCYL :GET THE CYLINDER ADDRESS  
18 014334 042737 170000 002304      BIC      #170000,DESCYL :REMOVE ANY STATUS BITS  
19 014342 062737 000002 002252      ADD      #2,TBLPTR     :GET THE TRACK ADDRESS  
20 014350 017737 165676 002244      MOV      @TBLPTR,DESTRK :AND LOAD IT NOW  
21 014356 105037 002244          CLRB     DESTRK       :EXTRACT SECTOR FROM DISK ADDRESS  
22 014362 062737 000014 002252      ADD      #<6*2>,TBLPTR :GET NEXT ENTRY IN BUFFER  
23 014370 000207          RTS      PC
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14 014372
15 014372 010046
16 014374 010146
17 014376 013746 002366
18 014402 004737 014452
19 014406 013700 002364
20 014412 062700 000225
21 014416 012701 000517
22 014422 005337 002366
23 014426 100402
24 014430 060100
25 014432 000773
26 014434 010037 002364
27 014440 012637 002366
28 014444 012601
29 014446 012600
30 014450 000207
31
32
33
34
35
36
37
38
39
40
41 014452 105737 002237
42 014456 001420
43 014460 010146
44 014462 010246
45 014464 113701 002307
46 014470 013702 002366
47 014474 006301
48 014476 006302
49 014500 020201
50 014502 103402
51 014504 162702 000061
52 014510 010237 002366

.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, ECCWRD
:*****
:OUTPUTS ARE: ECCWRD
:*****
: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 ECCWRD,RO ;:GET THE USER ECC INPUT
ADD #<G1+TD+G2+HDR0+G3>,RO ;:GET INFO TO PUT THE DEFECT
;:AT LEAST IN SECTOR 0
1\$: MOV #<HDR+G3+DATA+G4>,R1 ;:GET SIZE OF SECTOR
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,ECCWRD ;: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

:*****
:THIS MODULE IS USED WHEN A USER INPUTS A DEFECT VIA THE SECTOR WORD COUNT
:MECHANISM AND THE DRIVE IS IN INTERLEAVED MODE. THE DATA IS CONVERTED FROM
:INTERLEAVED POSITION TO LINEAR (NON-INTERLEAVED) POSITION.
:*****
:INPUTS ARE: SECADD
:*****
:OUTPUTS ARE: SECADD
:*****
:THIS MODULE IS CALLED BY: RELTIVE
:*****

MAPSEC: TSTB INTLEV ;:INTERLEAVED SECTORS ?
BEQ 2\$;:BR IF NO
MOV R1,-(SP) ;:PUSH R1 ON STACK
MOV R2,-(SP) ;:PUSH R2 ON STACK
MOVB PLTRK,R1 ;:GET THE FIRST INTERLEAVED SECTOR
MOV SECADD,R2 ;:SAVE THE SECTOR ADDRESS
ASL R1 ;:MULTIPLY BY TWO
ASL R2 ;:FOR THE THRESHOLD AND THE INPUT
CMP R2,R1 ;:SCALE THE USER INPUT
BLO 1\$;:IT JUST NEEDS DOUBLING, IF LOWER!
SUB #49,R2 ;:THIS IS A SECTOR ADDRESS => 25.
1\$: MOV R2,SECADD ;:SAVE THE CORRECTED SECTOR ADDRESS

52 014514 012602
014516 012601
53 014520 000207

2\$: MOV (SP)+,R2
MOV (SP)+,R1
RTS PC
::POP STACK INTO R2
::POP STACK INTO R1
: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
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

.SBTTL INSERT ABSOLUTE DEFECT INTO TDR

```

:*****
:THIS MODULE INSERTS THE ABSOLUTE DATA FROM RELATIVE TO INDEX INTO THE
:EXISTING TD RECORD (IF POSSIBLE). IF THE SCANNER IS BEING USED AND THE
:NUMBER OF DEFECTS EXCEEDS 4, THE USER BAD SECTOR FILE WILL AUTOMATICALLY
:BE UPDATED, ELSE AN ERROR MESSAGE WILL BE GENERATED.
:
:INPUTS ARE:  ECCWRD, TDWRD3 - TDWRD6
:
:OUTPUTS ARE: ECCWRD, TDWRD3 - TDWRD6
:
:THIS MODULE IS CALLED BY: INPUTD, HDSCAN, DASCAN, UPDSCR
:*****
  
```

INSERT:

```

MOV  R0,-(SP)      ;;PUSH R0 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$:  CMP  (R1)+,#NULL ;;IS THE TD RECORD FULL ?
     BEQ  2$        ;;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$:  MOV  #-1,TMPBSF ;;FLAG THAT DATA IS FOR TD, NOT BSF
     TST TDWRD3     ;;IS DATA VALID ?
     BEQ  9$        ;;BR IF NO
     MOV  #4,R0     ;;GET THE # OF DEFECT WORDS TO CHECK IN TD
     MOV  #TDWRD3,R1 ;;GET POINTER TO BEGINNING OF DEFECTS
     ADD  #<DS/2>,ECCWRD ;;ADD HALF A DEFECT TO ABSOLUTE VALUE
3$:  CMP  ECCWRD,(R1)+ ;;DOES NEW DEFECT GO BEFORE THIS OLD DEFECT ?
     BLO  5$        ;;BR IF YES
     SUB  -2(R1),ECCWRD ;;MAKE NEW DEFECT RELATIVE TO PREVIOUS DEFECT
     ADD  #<DS>,ECCWRD ;;ADD A DEFECT SKIP TO THE NEW DEFECT
     DEC  R0        ;;DONE LOOKING YET ?
     BGT  3$        ;;BR IF NO
     ;;SETUP TO PUT THIS NEW DEFECT INTO THE USR BSF
     ;;BUFFER,
4$:  MOV  DESCYL,TMPBSF ;;GET CYLINDER ADDRESS,
     MOV  SECADD,TMPBSF+2 ;;SECTOR ADDRESS AND
     MOVB DESTRK+1,TMPBSF+3 ;;TRACK ADDRESS.
     TST  SCANR      ;;ARE WE DOING PACK SCAN ?
     BNE  9$        ;;BR IF YES
     ;;PRINT '** WARNING ** TD RECORD IS FULL, INPUT
     ;;      DATA REJECTED'

MOV  #WARN,-(SP)
MOV  #FRMT60,-(SP)
MOV  #2,-(SP)
MOV  SP,R0
TRAP C$PNTF
ADD  #6,SP
BR   9$

5$:  SUB  #<DS/2>,ECCWRD ;;SUBTRACT A HALF DEFECT FROM ABSOLUTE VALUE
     TST -(R1)        ;;BACK THE POINTER UP BY ONE WORD
  
```

```

014522 010046
014522 010146
014524 010146
014526 012700 000004
014532 012701 030204
014536 022127 140000
014542 001403
014544 005300
014546 003373
014550 000430
014552 012737 177777 030244
014560 005737 030204
014564 001510
014566 012700 000004
014572 012701 030204
014576 062737 000066 002364
014604 023721 002364
014610 103437
014612 166137 177776 002364
014620 062737 000154 002364
014626 005300
014630 003365
014632 013737 002304 030244
014640 013737 002366 030246
014646 113737 002245 030247
014654 005737 002170
014660 001052
014662 012746 006016
014666 012746 005005
014672 012746 000002
014676 010600
014700 104417
014702 062706 000006
014706 000437
014710 162737 000066 002364
014716 005741
  
```


51	014720	020027	000001			CMP	RO,#1	::INSERT AFTER THREE DEFECTS?
52	014724	001416				BEQ	8\$::IF MATCH, YES
53	014726	020027	000002			CMP	RO,#2	::INSERT AFTER TWO DEFECTS?
54	014732	001411				BEQ	7\$::IF MATCH, YES
55	014734	020027	000003			CMP	RO,#3	::INSERT AFTER ONE DEFECT?
56	014740	001403				BEQ	6\$::IF MATCH, YES
57	014742	016161	000004	000006		MOV	4(R1),6(R1)	::SHUFFLE THE DATA DOWN
58	014750	016161	000002	000004	6\$:	MOV	2(R1),4(R1)	::TO MAKE ROOM FOR THE NEW DEFECT
59	014756	011161	000002		7\$:	MOV	(R1),2(R1)	::KEEP SHUFFLING DOWN
60	014762	013711	002364		8\$:	MOV	ECCWRD,(R1)	::LOG THE NEW DEFECT DATA NOW!
61	014766	005300				DEC	RO	::WAS THIS LAST ENTRY IN TD ?
62	014770	001406				BEQ	9\$::BR IF YES
63	014772	026127	000002	140000		CMP	2(R1),#NULL	::NEXT WORD NULL ?
64	015000	001402				BEQ	9\$::BR IF YES
65	015002	161161	000002			SUB	(R1),2(R1)	::ADJUST NEXT DEFECT TO MAKE IT RELATIVE TO
66								::THE NEW DEFECT
67	015006				9\$:			
	015006	012601				MOV	(SP)+,R1	::POP STACK INTO R1
	015010	012600				MOV	(SP)+,R0	::POP STACK INTO R0
68	015012	000207				RTS	PC	::NOW 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

.SBTTL TRACK DESCRIPTOR RECORD DUMP

```

:*****
:THIS MODULE PRODUCES THE TRACK DESCRIPTOR FILE ON THE LOCAL CONSOLE PRINTER.
:IT IS INVOKED ANYTIME A USER REQUESTS A TRACK DESCRIPTOR DUMP.
:
:INPUTS ARE: TEXT, TDWRD1 - TDWRD6
:
:OUTPUTS ARE: NONE
:
:THIS MODULE IS CALLED BY: INPUTD, READTD
:*****
    
```

TDDUMP:

```

MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV R2,-(SP)      ;;PUSH R2 ON STACK
TST LIST          ;;IS LIST OPTION ENABLED ?
BEQ 1$            ;BR IF NO
CMP DEF CNT,#1    ;WAS FIRST DEFECT LISTED ALL READY ?
BGE 2$            ;BR IF YES
1$:               ;PRINT THE APPROPRIATE MESSAGE
MOV TEXT,-(SP)
MOV #FRMT1,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #6,SP
                ;PRINT 'WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5
MOV #FRMT35,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP
2$:             MOV TDWRD2,-(SP)      ;GET THE 2ND WORD OF THE TD (TRK/SEC ADRS)
                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 DATA.
                MOV TDWRD1,-(SP)    ;GET THE FIRST WORD OF TD (CYLINDER ADDRESS)
                MOV (SP),-(SP)      ;SAVE 1ST TD WORD
                BIC #170000,2(SP)    ;STRIP THE STATUS BITS
                ;PRINT TD WORD#1, CYL, TD WORD#2 & TRK/SEC DATA
MOV (SP)+,-(SP)
MOV (SP)+,-(SP)
MOV (SP)+,-(SP)
MOV (SP)+,-(SP)
MOV (SP)+,-(SP)
MOV #FRMT36,-(SP)
MOV #6,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #16,SP
MOV #TDWRD3,R1    ;GET THE DEFECT FILE
MOV #4,R2          ;AND THE MAXIMUM NUMBER OF DEFECTS
3$:               CMP (R1)+,#NULL    ;IS IT NULL DATA ?
                BEQ 4$              ;BR IF YES
    
```

```

015014 010146
015014 010246
015016 010246
15 015020 005737 002200
16 015024 001404
17 015026 023727 002246 000001
18 015034 002022
19 015036
20 015036 013746 002344
    015042 012746 002645
    015046 012746 000002
    015052 010600
    015054 104417
    015056 062706 000006
21
22 015062 012746 004421
    015066 012746 000001
    015072 010600
    015074 104417
    015076 062706 000004
23 015102 013746 030202
24 015106 011646
25 015110 105066 000003
26 015114 011646
27 015116 105066 000002
28 015122 000366 000002
29 015126 013746 030200
30 015132 011646
31 015134 042766 170000 000002
32
33 015142 012646
    015144 012646
    015146 012646
    015150 012646
    015152 012646
    015154 012746 004530
    015160 012746 000006
    015164 010600
    015166 104417
    015170 062706 000016
34 015174 012701 030204
35 015200 012702 000004
36 015204 022127 140000
37 015210 001413
    
```



```

38
39 015212 016146 177776      MOV      -2(R1),-(SP)      ;PRINT THE DEFECT DATA
   015216 012746 003624      MOV      #FRMT14, -(SP)
   015222 012746 000002      MOV      #2, -(SP)
   015226 010600              MOV      SP, R0
   015230 104417              TRAP     C$PNTF
   015232 062706 000006      ADD      #6, SP
40 015236 000412              BR       5$
41 015240                    4$:
42 015240 016146 177776      MOV      -2(R1), -(SP)      ;PRINT ' 140000 '
   015244 012746 003635      MOV      #FRMT15, -(SP)
   015250 012746 000002      MOV      #2, -(SP)
   015254 010600              MOV      SP, R0
   015256 104417              TRAP     C$PNTF
   015260 062706 000006      ADD      #6, SP
43 015264 005302              5$:
44 015266 003346              DEC      R2
45                                BGT     3$
46 015270 012746 002642      MOV      #CRLF, -(SP)
   015274 012746 000001      MOV      #1, -(SP)
   015300 010600              MOV      SP, R0
   015302 104417              TRAP     C$PNTF
   015304 062706 000004      ADD      #4, SP
47 015310 012602              MOV      (SP)+, R2          ::POP STACK INTO R2
   015312 012601              MOV      (SP)+, R1          ::POP STACK INTO R1
48 015314 000207              RTS      PC                ;TAKE THE RETURN
  
```

```
1  
2  
3  
4  
5  
6  
7 015316 012702 031260 SESORT: MOV #USRBUF+8.,R2 :GET POINTER TO BEGINNING OF USER BUFFER  
8  
9 015322 010201 1$: MOV R2,R1 :TO GET OVER 2 S/N WRDS AND 2 ALL 0'S WRDS.  
10 015324 022121 :START WITH THE SAME BUFFER ADDRESS  
11 015326 022221 2$: CMP (R1)+,(R1)+ :GET OVER 1ST CYLINDER/TRACK/SECTOR WORDS  
12 015330 001017 :DUPLICATE DATA? (CYLINDER ADDRESS)  
13 015332 022221 BNE 5$ :BR IF NO  
14 015334 001014 :DUPLICATE DATA? (TRACK/SECTOR ADDRESS)  
15 015336 012161 177772 3$: MOV (R2)+,(R1)+ :BR IF NO  
16 015342 012161 177772 :STRIP THE DUPLICATE ENTRY AND PUSH DOWN  
17 015346 020127 032246 :KEEP PUSHING DOWN THE LIST  
18 015352 101771 :END OF USR BUFFER YET ?  
19 015354 012741 177777 :BR IF NO  
20 015360 012741 177777 :NULL THIS ENTRY (LAST ENTRY IN BUFFER)  
21 015364 000756 :AND THE PREVIOUS ENTRY  
22 :AND GO-ON!  
23 015366 024241 4$: CMP -(R2),-(R1) :PUSH THE BUFFER POINTERS (BACK TO THE TRACK ADDRESS  
24 015370 024221 5$: CMP -(R2),(R1)+ :RESTORE R2 AND POP R1 TO THE NEXT BUFFER ENTRY  
25 015372 020127 032246 :DONE WITH THIS COMPARISON?  
26 015376 103753 :BR IF NO  
27 015400 022222 :GET NEXT ENTRY TO LOOK FOR DUPLICATES  
28 015402 005712 :IS IT END OF BUFFER ?  
29 015404 100403 :BR IF NO  
30 015406 020227 032242 :DONE WITH ENTIRE BSF YET ?  
31 015412 103743 :BR IF NO  
32 015414 000207 6$: BLO 1$ :RETURN TO THE USER  
 :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
54
55
56
57

.SBTTL TRACK DESCRIPTOR CALCULATION MODULE

 : THIS ROUTINE COMPUTES THE ACTUAL OFFSET OF THE TD WORD, AND DETERMINES IF THE
 : DEFECT LIES WITHIN THE CURRENT SECTOR. IF THE DEFECT DOES, THE APPROPRIATE
 : CALCULATION IS MAPPED AND PREPARED FOR TRANSFER TO THE DRIVE.

INPUTS ARE: TDWRD3 - TDWRD6, DS, PARTBL,
 SIZE OF SECTOR (HDR+G3+DATA+G4)

OUTPUTS ARE: TDSBLK, RO - R5, DEFSEC, DEFTRK, TDCPY1 - TDCPY4,
 DEF1 - DEF4, DEFCNT

THIS MODULE IS CALLED BY: MAKTRK

17	015416	005037	002374		CALCTD: CLR	DEFTRK	:RESET THE # OF DEFECTS PER TRACK COUNTER
18	015422	012705	033400		MOV	#TDSBLK,R5	:POINT TO THE TD OUTPUT BLOCK
19	015426	013737	002224	002226	MOV	ENDPTR,ENDTAB	:GET THE BUFFER SIZE LIMIT
20	015434	006337	002226		ASL	ENDTAB	:PUT THE LIMIT ON A WORD BOUNDARY
21	015440	062737	033400	002226	ADD	#TDSBLK,ENDTAB	:AND SET THE LIMIT
22	015446	012704	030214		1\$: MOV	#TDCPY1,R4	:RESTORE POINTER TO 1ST TD WORD
23	015452	013724	030204		MOV	TDWRD3,(R4)+	:SAVE TD WORD #3
24	015456	013724	030206		MOV	TDWRD4,(R4)+	:SAVE TD WORD #4
25	015462	013724	030210		MOV	TDWRD5,(R4)+	:SAVE TD WORD #5
26	015466	013724	030212		MOV	TDWRD6,(R4)+	:SAVE TD WORD #6
27	015472	012704	030214		MOV	#TDCPY1,R4	:RESTORE POINTER TO 1ST TD WORD
28	015476	022427	140000		CMP	(R4)+,#NULL	:ANY DEFECTS ON THIS TRACK ?
29	015502	001007			BNE	3\$:BR IF YES
30	015504				2\$:		:NO MORE DEFECTS ON THIS TRACK, SO NULL THE
31							:SKIP DISPLACEMENT WORDS IN EACH OF THE
32	015504	012725	140000		MOV	#NULL,(R5)+	:REMAINING SECTORS.
33	015510	020537	002226		CMP	R5,ENDTAB	:DONE WITH THE TABLE YET ?
34	015514	003773			BLE	2\$:BR IF NO
35	015516	000137	016246		JMP	17\$:EXIT, GO WRITE FORMAT FOR THIS TRACK
36							:NOW CORRECT EACH SKIP TO THE BEGINNING OF
37							:THE DEFECT, SO IT CAN BE INSERTED INTO IT'S
38							:APPROPRIATE SECTOR.
39	015522	005237	002374		3\$: INC	DEFTRK	:ACCUMULATE # OF DEFECTS ON THIS TRACK
40	015526	023727	002374	000004	CMP	DEFTRK,#4	:DID WE REACH LAST DEFECT ?
41	015534	103003			BHS	4\$:BR IF YES
42	015536	022427	140000		CMP	(R4)+,#NULL	:ANY MORE DEFECTS ?
43	015542	001367			BNE	3\$:BR IF YES
44							: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	015544	012704	030214		4\$: MOV	#TDCPY1,R4	:RESTORE POINTER TO 1ST TD WORD
48	015550	004737	016252		JSR	PC,TDCASE	:LOOK FOR A TD AND/OR HDR 0 MOVED
49	015554	021427	140000		CMP	(R4),#NULL	:ANY MORE DEFECTS ?
50	015560	001751			BEQ	2\$:BR IF NO
51							:NOW FIND WHICH SECTOR THE DEFECT SHOULD BE
52							:INSERTED INTO.
53	015562	162714	000066		5\$: SUB	#<DS/2>,(R4)	:CORRECT 1ST DEFECT TO THE BEGINNING OF DEFECT
54	015566	105037	002240		CLRB	SAMSEC	:RESET THE SAME SECTOR FLAG,
55	015572	005037	002372		CLR	DEFSEC	:THE # OF DEFECTS PER SECTOR COUNTER AND
56	015576	005037	016250		CLR	TMDLTA	:THE TEMP. ACCUMULATIVE DELTA COUNTER.
57	015602	012700	000004		MOV	#4,RO	:GET # OF ENTRIES

58	015606	012701	002554		MOV	#DELTA,R1		:GET POINTER TO BEGINNING OF DELTA TABLE
59	015612	005021		6\$:	CLR	(R1)+		:INITIALIZE THE TABLE
60	015614	005300			DEC	R0		:DONE YET ?
61	015616	003375			BGT	6\$:BR IF NO
62	015620	012701	002554		MOV	#DELTA,R1		:GET POINTER TO BEGINNING OF DELTA TABLE
63	015624	005714			TST	(R4)		:ANY MORE DEFECTS ON TRACK ?
64	015626	003726			BLE	2\$:BR IF NO
65	015630	012700	000517		MOV	#<HDR+G3+DATA+G4>,R0		:GET THE SIZE OF SECTOR
66	015634	160014			SUB	R0,(R4)		:IS THE DEFECT IN THIS SECTOR ?
67	015636	003166			BGT	16\$:BR IF NO
68	015640	060014			ADD	R0,(R4)		:YES, RESTORE DEFECT
69								:NOW FIND WHERE THE BEGINNING OF THE DEFECT
70								:FALLS WITHIN THIS PARTICULAR SECTOR
71	015642	012703	002322		MOV	#PARTBL,R3		:GET POINTER TO BEGINNING OF SECTOR PARTITION TABLE
72	015646	012302			MOV	(R3)+,R2		:GET THE 1ST PARTITION
73	015650	021402		7\$:	CMP	(R4),R2		:IS THE DEFECT IN THIS PART OF THE SECTOR ?
74	015652	101410			BLOS	9\$:BR IF YES
75	015654	021400			CMP	(R4),R0		:IS THIS DEFECT AT THE SECTOR LIMIT ?
76	015656	001402			BEQ	8\$:BR IF YES
77	015660	062302			ADD	(R3)+,R2		:EXTEND THE PARTITION RANGE IN THE SECTOR
78	015662	000772			BR	7\$:AND TRY AGAIN
79	015664	005314		8\$:	DEC	(R4)		:FOUND DEFECT AT END OF SECTOR, REDUCE BY ONE
80	015666	005237	002372		INC	DEFSEC		:ONE MORE DEFECT FOUND IN THIS SECTOR
81	015672	000425			BR	10\$:TO FACILLITATE RP07 MICROCODE TESTING
82								
83	015674	005237	002372	9\$:	INC	DEFSEC		:ONE MORE DEFECT FOUND IN THIS SECTOR
84	015700	105737	002240		TSTB	SAMSEC		:IS THIS THE 1ST DEFECT IN THIS SECTOR ?
85	015704	001420			BEQ	10\$:BR IF YES
86								:NOW SEE IF THE BEGINNING OF THIS DEFECT FALLS
87								:WITHIN THE NEXT MINIMUM DATA SEGMENT OF THIS
88								:SECTOR
89	015706	021427	000176		CMP	(R4),#<DS+MSEG>		:IS DEFECT WITHIN NEXT MINIMUM DATA SEGMENT ?
90	015712	101015			BHI	10\$:BR IF NO
91	015714	012746	000176		MOV	#<DS+MSEG>,-(SP)		:::PUSH #<DS+MSEG> ON STACK
92	015720	011615			MOV	(SP),(R5)		:SAVE THIS VALUE IN THE DEFECT TABLE
93	015722	161416			SUB	(R4),(SP)		:GET DELTA FOR SKEW FORWARD TO THE MINIMUM DATA SEGMENT
94	015724	012621			MOV	(SP)+,(R1)+		:SAVE THE VALUE IN THE DELTA TABLE
95	015726	026427	000002 140000		CMP	2(R4),#NULL		:ANY MORE DEFECT ?
96	015734	001431			BEQ	12\$:BR IF NO
97	015736	166164	177776 000002		SUB	-2(R1),2(R4)		:ADJUST THE NEXT DEFECT BY DELTA
98	015744	000425			BR	12\$:NOW, GO AND SET THE STATUS
99								
100	015746	011415		10\$:	MOV	(R4),(R5)		:LOG THIS NON-CRITICAL INFORMATION
101	015750	105737	002240		TSTB	SAMSEC		:IS THIS THE 1ST DEFECT IN THIS SECTOR ?
102	015754	001021			BNE	12\$:BR IF NO
103								:NOW SEE IF THE BEGINNING OF THIS DEFECT FALLS
104								:WITHIN THE HEADER OF THIS SECTOR
105	015756	021527	000010		CMP	(R5),#<HDR>		:IS THE DEFECT WITHIN THE HEADER ?
106	015762	101014			BHI	11\$:BR IF NO
107	015764	012746	000010		MOV	#<HDR>,-(SP)		:::PUSH #<HDR> ON STACK
108	015770	161516			SUB	(R5),(SP)		:FIND THE SKEW VALUE AND
109	015772	061615			ADD	(SP),(R5)		:CORRECT THIS DEFECT VALUE
110	015774	012621			MOV	(SP)+,(R1)+		:SAVE THE VALUE IN THE DELTA TABLE
111	015776	026427	000002 140000		CMP	2(R4),#NULL		:ANY MORE DEFECT ?
112	016004	001403			BEQ	11\$:BR IF NO
113	016006	166164	177776 000002		SUB	-2(R1),2(R4)		:ADJUST THE NEXT DEFECT BY DELTA
114	016014	162715	000006	11\$:	SUB	#<HDR-2>,(R5)		:NOW MAKE THE REFERENCE TO THE BEGINNING

TRACK DESCRIPTOR CALCULATION MODULE

115									:OF THE 1ST HEADER CRC WORD
116									:AND POINT TO THE NEXT DEFECT ENTRY
117	016020	005725			12\$:	TST	(R5)+		:MOVE THE OUTPUT POINTER UP ONCE!
118	016022	004737	016400			JSR	PC,CHKSEG		:GO CHECK FOR MINIMUM DATA SEGMENT VIOLATIONS
119	016026	005337	002374			DEC	DEFTRK		:ONE LESS DEFECT TO GO ON THIS TRACK
120	016032	001624				BEQ	2\$:IF ZERO, TRACK RECORD IS COMPLETE
121	016034	013746	016250			MOV	TMDLTA,-(SP)		:PUSH TMDLTA ON STACK
122	016040	163716	002554			SUB	DELTA,(SP)		:ACCUMULATE ALL DELTA VALUES
123	016044	163716	002556			SUB	DELTA+2,(SP)		:SO THEY CAN BE SUBTRACTED FROM
124	016050	163716	002560			SUB	DELTA+4,(SP)		:THE BALANCE OF THE SECTOR REMAINS.
125	016054	163726	002562			SUB	DELTA+6,(SP)+		:ANY MORE DELTA'S ACCUMULATED ?
126	016060	002005				BGE	13\$:BR IF NO
127	016062	005446				NEG	-(SP)		:MAKE THE DIFFERENCE POSITIVE
128	016064	061637	016250			ADD	(SP),TMDLTA		:AND ADD IT TO OLD ACCUMULATED TEMP. DELTA
129	016070	161600				SUB	(SP),R0		:SUBTRACT DELTA FROM END OF SECTOR BOUNDRY AND
130	016072	162602				SUB	(SP)+,R2		:FROM SECTOR PARTITION VALUE.
131	016074	161400			13\$:	SUB	(R4),R0		:MAKE THE END OF SECTOR BOUNDRY RELATIVE TO
132	016076	062700	000154			ADD	#<DS>,R0		:THE PREVIOUS DEFECT AND ADD A SKIP TO EXTEND
133									:THE END OF SECTOR BOUNDRY.
134	016102	162402				SUB	(R4)+,R2		:MAKE THE SECTOR PARTITIONS RELATIVE TO THE
135	016104	062702	000154			ADD	#<DS>,R2		:PREVIOUS DEFECT AND ADD A SKIP TO EXTEND
136									:THE PARTITION RANGE.
137	016110	021427	000154			CMP	(R4),#<DS>		:WILL THIS DEFECT BE STACKED ?
138	016114	003406				BLE	14\$:BR IF YES
139	016116	021400				CMP	(R4),R0		:IS THE BEGINNING OF DEFECT IN CURRENT SECTOR ?
140	016120	003023				BGT	15\$:BR IF NO
141	016122	001660				BEQ	8\$:BR IF AT THE END OF THE CURRENT SECTOR
142	016124	105237	002240			INCB	SAMSEC		:MARK THIS DEFECT AS BEING IN SAME SECTOR
143	016130	000647				BR	7\$		
144									:NOW SETUP TO ADJUST THE SECTOR FOR STACKED
145									:DEFECTS
146	016132	005237	002372		14\$:	INC	DEFSEC		:ONE MORE DEFECT FOUND IN THIS SECTOR
147	016136	012715	000154			MOV	#<DS>,(R5)		:LOG THE STACKED DEFECT INFORMATION
148	016142	026427	000002	140000		CMP	2(R4),#NULL		:ANY MORE DEFECTS ?
149	016150	001723				BEQ	12\$:BR IF NO
150	016152	012746	000154			MOV	#<DS>,-(SP)		:VALUE OF STACKED SKIP
151	016156	161416				SUB	(R4),(SP)		:SUBTRACT CURRENT VALUE OF R4 FROM STACKED
152	016160	161664	000002			SUB	(SP),2(R4)		:VALUE AND ADJUST NEXT DEFECT BY DELTA
153	016164	012621				MOV	(SP)+,(R1)+		:LOG THE DELTA
154	016166	000714				BR	12\$		
155									
156	016170	160014			15\$:	SUB	R0,(R4)		:MODIFY THE SECTOR REMAINDER
157	016172	006337	002372			ASL	DEFSEC		:ADJUST DEFECT COUNT X 2
158	016176	006337	002372			ASL	DEFSEC		:ADJUST DEFECT COUNT X 4
159	016202	062737	016214	002372		ADD	#16\$,DEFSEC		:NOW FIND HOW MANY DEFECTS TO NULL OUT IN THE
160	016210	000177	164156			JMP	@DEFSEC		:REMAINDER OF THE HEADER WORDS IN THIS SECTOR
161									
162	016214	012725	140000		16\$:	MOV	#NULL,(R5)+		:NULL SD1.
163	016220	012725	140000			MOV	#NULL,(R5)+		:NULL SD2.
164	016224	012725	140000			MOV	#NULL,(R5)+		:NULL SD3.
165	016230	012725	140000			MOV	#NULL,(R5)+		:NULL SD4 IN THE HEADER OF THIS SECTOR
166	016234	020537	002226			CMP	R5,ENDTAB		:DID WE DO THE ENTIRE TRACK YET ?
167	016240	101002				BHI	17\$:BR IF YES
168	016242	000137	015566			JMP	5\$:AND CONTINUE
169	016246	000207			17\$:	RTS	PC		:EXIT, GO WRITE FORMAT FOR THIS TRACK
170									
171	016250	000000			TMDLTA:	.WORD	0		:TEMP. ACCUMULATIVE DELTA VALUE GOES HERE

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 ADJUST DEFECT TO TRACK DESCRIPTOR

 : THIS MODULE IS USED TO ADJUST ANY DEFECT DATA FOUND WHICH OCCURS SUBSEQUENT
 : TO A TD MOVED AND/OR A HDR 0 MOVED TRACK DESCRIPTOR VALUE. IT ALSO COUNTS
 : THESE DEFECTS IN THE OVERALL COUNTER.
 :*

: INPUTS ARE: BUFFER CALLED 'TDCPY1' (WHICH IS POINTED TO BY R4)

: OUTPUTS ARE: DEFSEC, BUFFER CALLED 'TDCPY1' (WHICH IS POINTED TO BY R4)

: CALL

JSR PC,TDCASE

16 016252 012700 002336
 17 016256 012702 000055
 18 016262 021420
 19 016264 003416
 20 016266 021420
 21 016270 003403
 22 016272 012702 000155
 23 016276 000436
 24
 25 016300 005337 002374
 26 016304 162702 000055
 27 016310 016046 177776
 28 016314 162416
 29 016316 062602
 30 016320 000414
 31
 32 016322 005337 002374
 33 016326 016446 000002
 34 016332 016046 177776
 35 016336 162416
 36 016340 061602
 37 016342 162616
 38 016344 005720
 39 016346 022620
 40 016350 003753
 41
 42 016352 021427 140000
 43 016356 001407
 44 016360 032737 140000 030200
 45 016366 001402
 46 016370 062702 000066
 47 016374 160214
 48 016376 000207

TDCASE: MOV #TDLMTS,R0 ;GET POINTER TO TD DEFECT LIMITS
 MOV #<TD+G2>,R2 ;INCLUDE THIS IN THE CALCULATION
 CMP (R4),(R0)+ ;WAS TD MOVED ?
 BLE 2\$;BR IF YES
 CMP (R4),(R0)+ ;WAS HDR0 MOVED ?
 BLE 1\$;BR IF YES
 MOV #<G1+TD+G2>,R2 ;INCLUDE THIS IN THE CALCULATION
 BR 4\$
 1\$: DEC DEFTRK ;ACCOUNT FOR HEADER 0 MOVED
 SUB #<TD+G2>,R2 ;ONE LESS DEFECT TO GO ON THIS TRACK
 MOV -2(R0),-(SP) ;INCLUDE THIS IN THE CALCULATION
 SUB (R4)+,(SP) ;GET HDR 0 MOVED LIMIT
 ADD (SP)+,R2 ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
 BR 3\$;AND ADD DIFFERENCE TO THIS CALCULATION.
 2\$: DEC DEFTRK ;ACCOUNT FOR TRACK DESCRIPTOR MOVED
 MOV 2(R4),-(SP) ;ONE LESS DEFECT TO GO ON THIS TRACK
 MOV -2(R0),-(SP) ;SAVE NEXT DEFECT VALUE
 SUB (R4)+,(SP) ;GET TD MOVED LIMIT
 ADD (SP),R2 ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
 SUB (SP)+,(SP) ;AND ADD DIFFERENCE TO THIS CALCULATION.
 TST (R0)+ ;ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
 CMP (SP)+,(R0)+ ;UPDATE LIMIT POINTER TO LOOK FOR HDR 0 MOVED
 BLE 1\$;WAS HDR 0 ALSO MOVED ?
 ;BR IF YES
 3\$: CMP (R4),#NULL ;ANY MORE DEFECTS ?
 BEQ 5\$;BR IF NO
 BIT #BIT14!BIT15,TDWRD1 ;WERE TD AND/OR HRD 0 REALLY MOVED ?
 BEQ 4\$;BR IF NO
 ADD #<DS/2>,R2 ;INCLUDE HALF A SKIP IN THE CALCULATION
 4\$: SUB R2,(R4) ;SUBTRACT ALONG WITH SKIP DEFECT INFO
 5\$: 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
54
55
56

```

:*****
:THIS MODULE IS USED TO CHECK FOR ANY MINIMUM DATA SEGMENT VIOLATIONS
:PRODUCED BY 'CALCTD'.
:
:INPUTS ARE: FROM (R5), WHICH IS A BUFFER CALLED 'TDSBLK', FROM (R1),
:            WHICH IS A BUFFER CALLED 'DELTA'. CONSTANTS USED ARE THOSE
:            WHICH DETERMINE A LEADING OR TRAILING DATA SEGMENT DEFECT
:            SKIP VIOLATION.
:
:OUTPUTS ARE: TO (R5), WHICH IS A BUFFER CALLED 'TDSBLK', TO (R1), WHICH
:            IS A BUFFER CALLED 'DELTA', TO (R4), WHICH IS A BUFFER
:            CALLED 'TDCPY1'.
:
:THIS MODULE IS CALLED BY: CALCTD
:*****
    
```

```

17 016400 010537 002214  CHKSEGE: MOV R5,TEMPA ;SAVE POINTER TO THIS DEFECT
18 016404 010046          MOV R0,-(SP) ;:PUSH R0 ON STACK
    016406 010246          MOV R2,-(SP) ;:PUSH R2 ON STACK
19 016410 005000          CLR R0 ;:RESET R0 (SEGMENT LEADING VS TRAILING)
20 016412 012701 002554  MOV #DELTA,R1 ;:GET POINTER TO BEGINNING TO DELTA TABLE
21 016416 013746 002372  MOV DEFSEC,-(SP) ;:GET THE # OF DEFECTS IN THIS SECTOR
22 016422 005745          1$: TST -(R5) ;:BACK UP THE DEFECT POINTER FOR THIS SECTOR
23 016424 005316          DEC (SP) ;:DONE ALL DEFECTS FOR THIS SECTOR YET ?
24 016426 003375          BGT 1$ ;:BR IF NO
25 016430 005726          TST (SP)+ ;:RESTORE THE STACK
26                                     ;:SETUP THE LEADING MINIMUM DATA SEGMENT VALUES
27 016432 012737 000042 002376  MOV #34.,MINSEG ;:MIN. PARTITION FOR DATA SEGMENT
28 016440 012737 000064 002400  MOV #52.,MAXSEG ;:MAX. PARTITION FOR DATA SEGMENT
29 016446 011502          MOV (R5),R2 ;:GET BEGINNING OF DEFECT
30                                     ;:NOW LOOK TO SEE IF THE BEGINNING OF THE DEFECT
31                                     ;:FALLS WITHIN GAP 3, BUT DON'T ALLOW THE DEFECT
32                                     ;:SKIP TO START ON THE LAST WORD OF GAP 3
33 016450 020237 002376          CMP R2,MINSEG ;:IS BEGINNING OF DEFECT IN GAP 3 ?
34 016454 103425          BLO 5$ ;:BR IF YES
35 016456 000404          BR 4$
36 016460 061502          2$: ADD (R5),R2 ;:ADD BEGINNING OF DEFECT TO CALCULATION
37                                     ;:NOW LOOK TO SEE IF THE BEGINNING OF THE
38                                     ;:DEFECT FALLS WITHIN THE MINIMUM DATA SEGMENT
39 016462 020237 002376          3$: CMP R2,MINSEG ;:IS BEGINNING OF DEFECT IN MINIMUM DATA SEGMENT ?
40 016466 101420          BLOS 5$ ;:BR IF NO
41 016470 020237 002400          4$: CMP R2,MAXSEG ;:IS BEGINNING OF DEFECT IN MINIMUM DATA SEGMENT ?
42 016474 101030          BHI 6$ ;:BR IF NO
43 016476 005711          TST (R1) ;:WAS THIS DEFECT PREVIOUSLY ADJUSTED ?
44 016500 001013          BNE 5$ ;:BR IF YES
45 016502 013746 002400          MOV MAXSEG,-(SP) ;:PUSH MAXSEG ON STACK
46 016506 160216          SUB R2,(SP) ;:FIND THE SKEW VALUE AND
47 016510 061615          ADD (SP),(R5) ;:CORRECT THIS DEFECT VALUE
48 016512 012611          MOV (SP)+,(R1) ;:SAVE THE VALUE IN THE DELTA TABLE
49 016514 026427 000002 140000  CMP 2(R4),#NULL ;:ANY MORE DEFECTS ?
50 016522 001402          BEQ 5$ ;:BR IF NO
51 016524 161164 000002          SUB (R1),2(R4) ;:REMOVE THE DELTA VALUE FROM NEXT DEFECT
52
53 016530 062737 000154 002376  5$: ADD #<DS>,MINSEG ;:ADD A DEFECT SKIP TO THE MIN. SEGMENT VALUE
54 016536 062737 000154 002400  ADD #<DS>,MAXSEG ;:AND A DEFECT SKIP TO THE MAX. SEGMENT VALUE
55 016544 022521          CMP (R5)+,(R1)+ ;:POP THE DATA AND DELTA FILES
56 016546 020537 002214          CMP R5,TEMPA ;:DONE CHECKING DATA SEGMENTS YET ?
    
```

57	016552	001413			BEQ	7\$:BR IF YES
58	016554	000741			BR	2\$:ELSE KEEP GOING...
59								
60	016556	005700			TST	R0	6\$:	:HAVE WE LOOKED AT THE TRAILING DATA SEGMENT YET ?
61	016560	100763			BMI	5\$:BR IF YES
62								:SETUP THE TRAILING MINIMUM DATA SEGMENT VALUES
63	016562	062737	000360	002376	ADD	#240.,MINSEG		:MIN. PARTITION FOR DATA SEGMENT
64	016570	062737	000360	002400	ADD	#240.,MAXSEG		:MAX. PARTITION FOR DATA SEGMENT
65	016576	005100			COM	R0		:MARK THIS EVENT
66	016600	000730			BR	3\$:TRY AGAIN...
67	016602						7\$:	
	016602	012602			MOV	(SP)+,R2		::POP STACK INTO R2
	016604	012600			MOV	(SP)+,R0		::POP STACK INTO R0
68	016606	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
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
55
56
57

.SBTTL FILE MERGE MODULE FOR TRACK RECORD

```

:*****
:THIS MODULE CREATES THE TRACK RECORD WHICH WILL BECOME THE FORMAT ON AN RP07,
:FOR A GIVEN TRACK.
:
:INPUTS ARE: INTLEV, PLTRK, TKWCNT, DESCYL, DESTRK, TDSBLK
:
:OUTPUTS ARE: HDRBLK+0 - HDRBLK+300
:
:THIS MODULE IS CALLED BY: MAKTRK
:*****
  
```

```

MERGE: CLR R1 ;CLEAN JUNK AS THESE REGISTERS
        CLR R2 ;ARE FLAGS FOR THE REST OF THE ROUTINE
        MOV #TDSBLK,R5 ;GET THE INITIAL TD TABLE ADDRESS
        TSTB INTLEV ;IS THE INTERLEAVE SWITCH ON?
        BEQ 1$ ;BR IF NO
        COMB R2 ;R2 IS THE FAST MASK
1$: MOV #PLTRK,R0 ;GET THE RP07 INTERLEAVE MASK
     MOVB (R0)+,R3 ;GET THE SECTOR COUNT
     MOVB (R0),R4 ;NOW GET THE HIGH OR INTERLEAVED INFO
     MOV #HDRBLK,R0 ;GET THE TABLE ADDRESS
     MOV TKWCNT,ENDTAB ;GET THE BUFFER LIMIT (6 WRDS X 50. SECTORS)
     ASL ENDTAB ;ADJUST IT FOR A WORD COUNT (X 2)
     ADD #HDRBLK,ENDTAB ;AND SET IT
     MOV #TDSBLK,R5 ;GET THE TD FILE ADDRESS
2$: BIS #BIT15!BIT14!BIT12,DESCYL ;LOG THESE SECTORS AS GOOD
     BIS DESTRK,R4 ;SET THE TRACK ADDRESS
     BIS DESTRK,R3 ;FOR BOTH FILE FLAGS
3$: MOV DESCYL,(R0)+ ;LOAD THE CYLINDER ADDRESS
     TSTB R1 ;WHICH SECTOR SHOULD BE LOADED?
     BEQ 4$ ;SECTOR FROM R4
     MOV R3,(R0)+ ;LOAD HIGH SECTOR
     INC R3 ;UP THE SECTOR COUNT
     CLRB R1 ;CLEAR TOGGLE POINTER
     BR 5$ ;AND PROGRESS FORWARD

4$: MOV R4,(R0)+ ;LOAD LOW OR NON-INTERLEAVED SECTOR
     INC R4 ;AND INCREMENT THE SECTOR ADDRESS
     TSTB R2 ;INTERLEAVE?
     BEQ 5$ ;NOT IF ZERO
     COMB R1 ;RESET THE TOGGLE POINTER
5$: MOV (R5)+,(R0)+ ;LOG THE TD DATA
     MOV (R5)+,(R0)+ ;ALL FOUR WORDS OF IT
     MOV (R5)+,(R0)+ ;FOR EACH SECTOR
     MOV (R5)+,(R0)+ ;THIS IS LAST WORD FOR EACH SECTOR
     CMP ENDTAB,R0 ;TABLE AT MAX?
     BLOS 7$ ;WRITE THE TD INFORMATION NOW!
     CMPB SEC50,R4 ;IS THE SECTOR ADDRESS FILE AT MAX?
     BHIS 6$ ;NOT YET, GO ON
     CLR R4 ;RESET R4 = TO SECTOR ZERO
     BR 2$ ;AND CONTINUE

6$: CMPB SEC50,R3 ;IS THIS SECTOR FILE AT MAX?
     BHIS 3$ ;NOT YET
     CLR R3 ;RESET R3 = SECTOR 0
  
```

```

14 016610 005001
15 016612 005002
16 016614 012705 033400
17 016620 105737 002237
18 016624 001401
19 016626 105102
20 016630 012700 002307
21 016634 112003
22 016636 111004
23 016640 012700 032250
24 016644 013737 002350 002226
25 016652 006337 002226
26 016656 062737 032250 002226
27 016664 012705 033400
28 016670 052737 150000 002304
29 016676 053704 002244
30 016702 053703 002244
31 016706 013720 002304
32 016712 105701
33 016714 001404
34 016716 010320
35 016720 005203
36 016722 105001
37 016724 000405
38
39 016726 010420
40 016730 005204
41 016732 105702
42 016734 001401
43 016736 105101
44 016740 012520
45 016742 012520
46 016744 012520
47 016746 012520
48 016750 023700 002226
49 016754 101412
50 016756 123704 002306
51 016762 103002
52 016764 005004
53 016766 000743
54
55 016770 123703 002306
56 016774 103344
57 016776 005003
  
```

58	017000	000736				BR	2\$;AND CONTINUE
59									
60	017002	042737	150000	002304	7\$:	BIC	#BIT15!BIT14!BIT12,DESCYL		;THESE BITS DON'T APPLY AS AN ADDRESS
61	017010	000207				RTS	PC		;AND TAKE THE RETURN

1
2
3
4
5
6
7
8
9
10
11
12
13
14 017012
017012 010146
017014 010246
017016 010346
15 017020 012702 002424
16 017024 012701 002500
17 017030 012703 000024
18 017034 013221
19 017036 005303
20 017040 003375
21 017042 005737 002416
22 017046 001402
23 017050 013221
24 017052 013221
25 017054
017054 012603
017056 012602
017060 012601
26 017062 000207

.SBTTL REGISTER BUFFER MODULE

```

:*****
:THIS MODULE CREATES 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 RH1i
      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
  
```

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

.SBTTL HEADER VERIFY MODULE

```

:*****
:THIS MODULE DOES THE HEADER AND S.D. VERIFICATION FOR A VERIFY MODE OF
:OPERATION. THE TRACK DESCRIPTOR IS READ, AND CALCULATIONS ARE PERFORMED
:ON THE TD WORDS FOUND. THESE CALCULATIONS ARE THEN TABLED AND COMPARED
:WITH THE ACTUAL S.D. WORDS FOUND ON A SPECIFIED TRACK. ANY MISMATCH WILL
:RESULT IN SEVERAL RETRY ATTEMPTS, THEN FINALLY AN ERROR MESSAGE.
:
:INPUTS ARE: HDRBLK+0 - HDRBLK+300., SEC50, INTLEV, RTYCNT, TDWRD3 - TDWRD6
:
:OUTPUTS ARE: DEFCNT, ERRMSK, DEF1 - DEF4, LINBUF+0 - LINBUF+300.
:
:THIS MODULE IS CALLED BY: MAIN
:*****
  
```

```

17 017064 012700 032250
18 017070 105737 002237
19 017074 001404
20 017076 004737 011264
21 017102 012700 034530
22 017106 113702 002306
23 017112 012701 033400
24 017116 012704 000006
25 017122 022021
26 017124 001007
27 017126 005304
28 017130 001374
29 017132 005302
30 017134 002370
31 017136 005037 002264
32
33 017142 000207
34
35 017144 014037 002270
36 017150 014137 002272
37 017154 012737 000010 002264
38
39 017162 000207
  
```

```

VERIFY: MOV      #HDRBLK,R0      ;GET POINTER TO BEGINNING OF EXPECTED DATA
        TSTB     INTLEV          ;INTERLEAVED DRIVE ?
        BEQ      1$              ;BR IF NO
        JSR      PC,CHABUF       ;RE-SHUFFLE THE DATA FOR A LINEAR COMPARE
        MOV      #LINBUF,R0      ;GET POINTER TO BEGINNING OF EXPECTED DATA
1$:     MOV      SEC50,R2         ;# OF SECTORS ON TRACK
        MOV      #TDSBLK,R1      ;GET POINTER TO BEGINNING OF RECEIVED DATA
2$:     MOV      #<HDR-2>,R4      ;# OF WORDS IN HEADER TO VERIFY
3$:     CMP      (R0)+,(R1)+      ;DOES EXPECTED DATA MATCH RECEIVED DATA ?
        BNE     4$              ;BR IF NO
        DEC     R4               ;DONE COMPARING THIS SECTOR YET ?
        BNE     3$              ;BR IF NO
        DEC     R2               ;DONE COMPARING THIS TRACK YET ?
        BGE     2$              ;BR IF NO
        CLR     ERRMSK          ;RESET NO ERROR STATUS AND SETUP CONITION CODE
        RTS     PC              ;IN PSW FOR RETURN

4$:     MOV      -(R0),EXPCD      ;SAVE THE EXPECTED DATA
        MOV      -(R1),RECVD     ;AND THE RECEIVED DATA
        MOV      #10,ERRMSK     ;LOG 'FORMAT VERIFY ERROR' STATUS AND
        RTS     PC              ;SETUP CONDITION CODE IN PSW FOR RETURN
  
```


.SBTTL READ BAD SECTOR FILE (DEC 144 FILE) MODULE

```

:*****
:THIS MODULE DOES A READ OF THE LAST USER CYLINDER/TRACK. IF THE READ IS
:UNSUCCESSFUL, (OPI = 1), THE PROGRAM ASSUMES THAT DEC STANDARD 144 WAS NEVER
:CREATED, AND MARKS STATUS AS SUCH. IF DATA IS FOUND FROM ONE OF SEVERAL
:SECTORS, KNOWN AS THE USER SECTOR FILE, THE INFORMATION IS STORED FOR A LATER
:REWRITE.
:
:INPUTS ARE:  USRBUF, RDAT, RTYCNT
:
:OUTPUTS ARE: SUPRSS, DESCYL, DESTRK, BADDR, FUNCTN, NEGWRD, ERRMAX, CREATE
:
:THIS MODULE IS CALLED BY: MAIN
:*****
    
```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 017164 012737 177777 002242 RDBSF:  MOV    #-1,SUPRSS      ;SUPPRESS THE ERROR MESSAGE OUTPUT
18 017172 010046                MOV    RO,-(SP)        ;;PUSH RO ON STACK
    017174 010146                MOV    R1,-(SP)        ;;PUSH R1 ON STACK
    017176 013746 002304        MOV    DESCYL,-(SP)    ;;PUSH DESCYL ON STACK
    017202 013746 002244        MOV    DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
    017206 013746 002370        MOV    FUNCTN,-(SP)    ;;PUSH FUNCTN ON STACK
19 017212 005000                CLR    RO              ;GET # USR BLOCKS DEC144 FILE
20 017214 012737 001165 002304    MOV    #629.,DESCYL    ;GET THE LAST CYLINDER,
21 017222 112737 000012 002244    MOVVB  #10.,DESTRK     ;THE FIRST USER SECTOR AND
22 017230 112737 000037 002245    MOVVB  #31.,DESTRK+1   ;THE LAST TRACK
23 017236 012737 031250 002230    MOV    #USRBUF,BADDR   ;GET POINTER TO BEGINNING OF USR BSF BUFFER
24 017244 012737 000071 002370    MOV    #RDDAT,FUNCTN   ;SET UP FOR A READ DATA COMMAND
25 017252 062737 000002 002244 1$:  ADD    #2,DESTRK       ;THE USER BAD SECTOR FILE
26 017260 012737 000400 002362 2$:  MOV    #256.,NEGWRD    ;WORD COUNT FOR THE ENTIRE SECTOR
27 017266 004737 010166                JSR    PC,DRIVER       ;EXECUTE THE OPERATION
28 017272 004737 012446                JSR    PC,RETRY        ;SEE IF WE HAD ANY ERRORS
29 017276 000404                BR     3$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
30                                ;ELSE RETURN HERE
31 017300 005737 002266                TST    RTYCNT          ;DID WE HAVE AN ERROR ?
32 017304 001365                BNE   2$              ;BR IF YES
33 017306 000422                BR     4$              ;GOOD READ!!!
34
35 017310 005200                3$:  INC    RO              ;INCREMENT TO NEXT COPY OF USR BSF
36 017312 020027 000004                CMP    RO,#4           ;DID WE TRY READING ALL COPIES OF USR BSF ?
37 017316 103755                BLO   1$              ;BR IF NO
38                                ;PRINT '** WARNING ** CORRUPT BAD SECTOR FILE,
39                                ;      WILL BE INITIALIZED'
40 017320 012746 006016                MOV    #WARN,-(SP)
    017324 012746 005327                MOV    #FRMT64,-(SP)
    017330 012746 000002                MOV    #2,-(SP)
    017334 010600                MOV    SP,RO
    017336 104417                TRAP  C$PNTF
    017340 062706 000006                ADD    #6,SP
41 017344 012700 031250                MOV    #USRBUF,RO     ;GET THE START OF THE USER DEC144 BSF
42 017350 004737 017664                JSR    PC,CRE144      ;CREATE THE USR BSF BUFFER
43                                ;NOW THAT WE HAVE A GOOD READ OR AN
44                                ;INITIALIZED BUFFER, LETS SEE IF THE DATA
45                                ;IS VALID.
46 017354 012701 031250                4$:  MOV    #USRBUF,R1     ;GET POINTER TO BEGINNING OF USER BSF
47 017360 022127 000105                CMP    (R1)+,#105     ;LOOK FOR THE PACK S.N.
48 017364 001351                BNE   3$              ;NO MATCH, REWRITE THE USER B.S.F
    
```

49	017366	022127	000105	CMP	(R1)+,#105	:LOOK FOR THE PACK S.N.
50	017372	001346		BNE	3\$:NO MATCH, REWRITE THE B.S.F
51	017374	005721		TST	(R1)+	:THIS WORD ALWAYS 0!
52	017376	001344		BNE	3\$:IF NOT, FILE IS BAD
53	017400	005721		TST	(R1)+	:IS THIS AN ALIGNMENT PACK ?
54	017402	001342		BNE	3\$:BR IF YES
55	017404	004737	012540	JSR	PC,RESET	:RESET ANY ERRORS
56	017410	012637	002370	MOV	(SP)+,FUNCTN	::POP STACK INTO FUNCTN
	017414	012637	002244	MOV	(SP)+,DESTRK	::POP STACK INTO DESTRK
	017420	012637	002304	MOV	(SP)+,DESCYL	::POP STACK INTO DESCYL
	017424	012601		MOV	(SP)+,R1	::POP STACK INTO R1
	017426	012600		MOV	(SP)+,R0	::POP STACK INTO R0
57	017430	005037	002242	CLR	SUPRSS	:ALLOW ERROR MESSAGE TO BE OUTPUT
58	017434	000207		RTS	PC	: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
48
49
50
51

.SBTTL WRITE BAD SECTOR FILE (DEC 144 FILE) MODULE

```

:*****
:THIS MODULE CREATES AND RESTORES (WRITES) THE BAD SECTOR FILE (DEC 144 FILE)
:ON THE LAST USER CYLINDER/TRACK OF THE DISK.
:
:INPUTS ARE:   WRTDAT, WRDAT, MFGBUF, USRBUF, CREATE
:
:OUTPUTS ARE:  DESCYL, DESTRK, FUNCTN, ERRFNC, MFG144, BADDR, NEGWRD, ERRMSK,
:              SEC50
:
:THIS MODULE IS CALLED BY: MAIN
:*****
    
```

WRTBSF:

```

MOV   DESCYL,-(SP)      ;;PUSH DESCYL ON STACK
MOV   DESTRK,-(SP)     ;;PUSH DESTRK ON STACK
MOV   #629.,DESCYL     ;;LOAD THE LAST CYLINDER,
MOVB  #31.,DESTRK+1    ;;THE LAST TRACK
CLRB  DESTRK           ;;AND SECTOR 0
MOV   #WRTDAT,FUNCTN   ;;SET UP FOR A WRITE DATA FUNCTION
MOV   #WRDAT,ERRFNC    ;;SAVE FUNCTION FOR ERROR ROUTINE
MOV   #MFGBUF,RO       ;;GET THE START OF THE MANUFACTURING DEC144 BSF
JSR   PC,CRE144        ;;CREATE THE MFG BSF BUFFER
MOV   #MFGBUF,BADDR    ;;GET POINTER TO BEGINNING OF MFG BSF BUFFER
MOV   #256.,NEGWRD     ;;AND THE SECTOR WORD COUNT
JSR   PC,DRIVER        ;;EXECUTE THE OPERATION
JSR   PC,RETRY         ;;SEE IF WE HAD ANY ERRORS
BR    2$              ;;RETURN HERE IF EXCEEDED RETRY LIMIT
                        ;;ELSE RETURN HERE
TST   RTYCNT           ;;DID WE HAVE AN ERROR ?
BNE   1$              ;;BR IF YES
INCB  DESTRK           ;;NEXT SECTOR
CMPB  DESTRK,#10.     ;;DONE WITH MFG FILE YET ?
BLO   1$              ;;BR IF NO
                        ;;START WRITTING USER 16 & 18 BIT BSF
MOV   #USRBUF,BADDR    ;;GET POINTER TO BEGINNING OF USR BSF BUFFER
MOV   #256.,NEGWRD     ;;WORD COUNT FOR SECTOR
JSR   PC,DRIVER        ;;EXECUTE THE OPERATION
JSR   PC,RETRY         ;;SEE IF WE HAD ANY ERRORS
BR    5$              ;;RETURN HERE IF EXCEEDED RETRY LIMIT
                        ;;ELSE RETURN HERE
TST   RTYCNT           ;;DID WE HAVE AN ERROR ?
BNE   4$              ;;BR IF YES
INCB  DESTRK           ;;NEXT SECTOR
CMPB  DESTRK,SEC50    ;;DONE WITH USR FILE YET ?
BHI   6$              ;;YES, TAKE RETURN
CMP   #MFGBUF,BADDR    ;;DID WE JUST WRITE A 18 BIT USER FILE ?
BEQ   3$              ;;BR IF YES
MOV   #MFGBUF,BADDR    ;;GET POINTER TO BEGINNING OF MFG BSF BUFFER
BR    4$              ;;GO WRITE A 18 BIT FILE
6$:  MOV   (SP)+,DESTRK  ;;POP STACK INTO DESTRK
      MOV   (SP)+,DESCYL ;;PCP STACK INTO DESCYL
      RTS   PC
    
```

```

017436 013746 002304
017436 013746 002244
017442 013746 002244
017446 012737 001165 002304
017454 112737 000037 002245
017462 105037 002244
017466 012737 000061 002370
017474 012737 005435 002254
017502 012700 030250
017506 004737 017664
017512 012737 030250 002230 1$:
017520 012737 000400 002362
017526 004737 010166
017532 004737 012446
017536 000403
017540 005737 002266
017544 001362
017546 105237 002244 000012 2$:
017552 123727 002244
017560 103754
017562 012737 031250 002230 3$:
017570 012737 000400 002362 4$:
017576 004737 010166
017602 004737 012446
017606 000403
017610 005737 002266
017614 001365
017616 105237 002244 002306 5$:
017622 123737 002244
017630 101010
017632 022737 030250 002230
017640 001750
017642 012737 030250 002230
017650 000747
017652
017652 012637 002244
017656 012637 002304
017662 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

```

:*****
: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:  MFGBUF OR USRBUF
:
: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
    
```

```

:*****
:THIS MODULE CONTROLS SPIRAL READS OR WRITES. IT INCREMENTS THE DESIRED TRACK
:ADDRESS UNTIL IT IS READY TO OVERFLOW INTO AN ILLEGAL ADDRESS. WHEN THIS IS
:READY TO HAPPEN, THE DESIRED TRACK IS RESET TO "MINTRK" AND THE DESIRED
:CYLINDER IS INCREMENTED. THE CALLING MODULE MUST DETECT WHEN THE DESIRED
:CYLINDER HAS OVERFLOWED.
:
:VARIABLES:   DESCYL AND DESTRK ARE AFFECTED BY THIS MODULE. ALSO, THIS
:              MODULE IS AFFECTED BY THE VALUE IN "MAXCYL" WHICH REPRESENTS
:              THE LAST CYLINDER ADDRESS ON THE DRIVE.
:
:INPUTS ARE:   MINTRK, MAXTRK, MAXCYL
:
:OUTPUTS ARE:  DESCYL, DESTRK
:
:THIS MODULE IS CALLED BY: MAIN
:*****
    
```

```

SPIRAL: INCB   DESTRK+1      ;INCREMENT TO NEXT TRACK ADDRESS
        CMPB   DESTRK+1,MAXTRK ;TRACK COUNT AT MAX?
        BLOS  1$            ;BR IF NO
        MOVB  MINTRK,DESTRK+1 ;RESTORE MIN TRACK ADDRESS
        INC   DESCYL         ;INCREMENT TO NEXT CYLINDER ADDRESS
1$:     RTS   PC            ;EXIT
    
```

```

017664 010001
017666 062701 001000
017672 012720 000105
017676 012720 000105
017702 005020
017704 005020
017706 012720 177777
017712 020001
017714 103774
017716 000207

105237 002245
123737 002245 002212
101405
113737 002210 002245
005237 002304
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

```

:*****
:THIS MODULE IS USED TO CONTROL THE COUNTING OF DEFECTS ENCOUNTERED ON THE
:RPO7 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
MOV TDWRD1,R1     ;;GET THE FIRST WORD OF TD (CYLINDER ADDRESS)
BIC #170000,R1    ;;STRIP THE STATUS BITS
MOV TDWRD2,R2     ;;GET THE SECOND WORD OF THE TD (TRK/SEC ADRS)
CLRB R2           ;;GET RID OF LOW BYTE (EXTRACT TRACK) AND
SWAB R2           ;;MAKE IT THE LOW BYTE DATA.
;;PRINT '** WARNING ** CYL XXX., TRK XX.
;;HAS 4 DEFECTS IN TD RECORD
MOV R2,-(SP)
MOV R1,-(SP)
MOV #WARN,-(SP)
MOV #FRMT61,-(SP)
MOV #4,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #12,SP
2$: INC -(R3)      ;;INCLUDE THIS IN # DEFECTS/TRACK
3$: MOV (SP)+,R3    ;;POP STACK INTO R3
MOV (SP)+,R2      ;;POP STACK INTO R2
MOV (SP)+,R1      ;;POP STACK INTO R1
RTS PL           ;;RETURN TO THE USER
    
```

```

017750
017750 010146
017752 010246
017754 010346
16 017756 023727 030204 140000
17 017764 001451
18 017766 005737 030204
19 017772 001446
20 017774 012702 000004
21 020000 012701 030204
22 020004 012703 002274
23 020010 022127 140000
24 020014 001434
25 020016 005237 002246
26 020022 005723
27 020024 005302
28 020026 003370
29 020030 005737 002200
30 020034 001024
31 020036 013701 030200
32 020042 042701 170000
33 020046 013702 030202
34 020052 105002
35 020054 000302
36
37
38 020056 010246
020060 010146
020062 012746 006016
020066 012746 005064
020072 012746 000004
020076 010600
020100 104417
020102 062706 000012
39 020106 005243
40 020110
020110 012603
020112 012602
020114 012601
41 020116 000207
    
```

```

1
2
3
4
5
6
7 020120 012737 001167 002304 FETWO: MOV #631.,DESCYL ;GET THE SECOND FE CYLINDER ADDRESS
8 020126 005037 002244 CLR DESTRK ;RESET THE TRACK/SECTOR ADDRESS
9 020132 012701 030200 1$: MOV #TDWRD1,R1 ;SET UP A BUFFER POINTER
10 020136 013711 002304 MOV DESTRK,(R1) ;FIRST WORD OF TD
11 020142 052721 030000 BIS #BIT13!BIT12,(R1)+ ;IT'S A TD - IN 16 BIT MODE
12 020146 013711 002244 MOV DESTRK,(R1) ;TRACK / SECTOR ADDRESS
13 020152 052721 000377 BIS #377,(R1)+ ;SECTOR ADDRESS = - 1
14 020156 012702 000004 MOV #4,R2 ;A RECORD ITERATION COUNTER
15 020162 012721 140000 2$: MOV #NULL,(R1)+ ;NO DATA FOR THIS TD (LEAVE IT NULL)
16 020166 005302 DEC R2 ;ONE LESS RECORD TO GO
17 020170 003374 BGT 2$ ;IF NOT ZERO, WE'RE NOT DONE
18 020172 004737 020222 JSR PC,WRITTD ;ATTEMPT TO WRITE THIS RECORD
19 020176 000207 RTS PC ;RETURN HERE IF EXCEEDED RETRY LIMIT
20 ;OTHERWISE RETURN HERE
21 020200 105237 002245 INCB DESTRK+1 ;NEXT TRACK PLEASE.....
22 020204 123727 002245 000037 CMPB DESTRK+1,#31. ;LAST TRACK YET?
23 020212 101747 BLOS 1$ ;TAKE BRANCH IF NOT
24 020214 105337 002245 DECB DESTRK+1 ;GET RID OF THE OVERFLOW ADDRESS
25 020220 000207 RTS PC ;TAKE THE USER 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
47
48
49
50
51

.SBTTL WRITE TRACK DESCRIPTOR MODULE

```

:*****
:THIS MODULE SIZES THE TD DATA AND DETERMINES IF THE TD ITSELF MUST BE
:RELOCATED DUE TO THE PRESENCE OF A DEFECT WHERE THE TD WOULD NORMALLY BE
:FOUND. IT ALSO DETERMINES IF HEADER #0 MUST BE RELOCATED DUE TO A DEFECT
:BEWEEN THE TD FILE AND THE FIRST HEADER. MASK BITS ARE SET TO INFORM THE
:DEVICE EITHER OF THESE CONDITIONS OCCUR.
:
:INPUTS ARE: TDWRD1, TDWRD3, TDWRD4, TDLMTS, WRITD, TDBCNT, RTYCNT
:
:OUTPUTS ARE: DESTRK, RTYCNT, FUNCTN, BADDR, ERRFNC, RPOF, NEGWRD, ERRMSK
:
:CALL
:      JSR    PC,WRITD      ;CALL WRITE TD SUBROUTINE
:      -----
:      -----              ;RETURN HERE IF EXCEEDED RETRY LIMIT
:      -----              ;ELSE RETURN HERE, NO ERROR
:
:THIS MODULE IS MAINLY CALLED BY: INPUTD, FETWO, MAIN
:*****
    
```

WRITD:

```

MOV    R0,-(SP)      ;:PUSH R0 ON STACK
MOV    R1,-(SP)      ;:PUSH R1 ON STACK
MOV    DESTRK,-(SP)  ;:PUSH DESTRK ON STACK
1$:    MOV    #TDLMTS,R0 ;:GET POINTER TO TD LIMITS
        MOV    #TDWRD3,R1 ;:GET TD RECORD POINTER
        BIC    #BIT14!BIT15,TDWRD1 ;:RESET TD & HDR0 MOVED BITS
        MOVB  #377,DESTRK ;:LOAD THE SECTOR ADDRESS BITS
        CMP   (R1),#NULL ;:ANY DEFECT SKIPS ON THIS TRACK ?
        BEQ   2$
        CMP   (R1),(R0)+ ;:WAS TD MOVED ?
        BLE  5$
        BLE  4$
        CMP   (R1),(R0)+ ;:WAS HDR 0 MOVED ?
        BLE  4$
2$:    MOV    #WRITD,FUNCTN ;:SET UP FOR A WRITE TD FUNCTION
        MOV    #TDWRD1,BADDR ;:GET POINTER TO BEGINNING OF TD RECORD BUFFER
        MOV    #MSWRD,ERRFNC ;:SAVE FUNCTION FOR ERROR ROUTINE
        BIS   #CMOD,@RPOF ;:SET COMMAND MODIFIER BIT
        MOV   TDBCNT,NEGWRD ;:6 WORD TRANSFER
        JSR   PC,DRIVER ;:SEND COMMAND TO CONTROLLER
        JSR   PC,RETRY ;:SEE IF WE HAD ANY ERRORS
        BR    3$ ;:RETURN HERE IF EXCEEDED RETRY LIMIT
        ;:ELSE RETURN HERE
        TST   RTYCNT ;:DID WE HAVE AN ERROR ?
        BNE  1$ ;:BR IF YES
        ADD  #2,6(SP) ;:ADJUST FOR GOOD RETURN
3$:    MOV    (SP)+,DESTRK ;:POP STACK INTO DESTRK
        MOV    (SP)+,R1 ;:POP STACK INTO R1
        MOV    (SP)+,R0 ;:POP STACK INTO R0
        RTS   PC ;:RETURN
4$:    BIS   #BIT15,TDWRD1 ;:MARK HDR 0 MOVED
        BR    2$ ;:EXECUTE FUNCTION NOW
5$:    BIS   #MTD,@RPOF ;:MARK TD MOVED IN OFFSET REG
    
```

```

020222
020222 010046
020224 010146
020226 013746 002244
23 020232 012700 002336
24 020236 012701 030204
25 020242 042737 140000 030200
26 020250 112737 000377 002244
27 020256 021127 140000
28 020262 001404
29 020264 021120
30 020266 003445
31 020270 021120
32 020272 003437
33 020274 012737 000065 002370
34 020302 012737 030200 002230
35 020310 012737 005414 002254
36 020316 052777 100000 162132
37 020324 013737 002314 002362
38 020332 004737 010166
39 020336 004737 012446
40 020342 000406
41
42 020344 005737 002266
43 020350 001330
44 020352 062766 000002 000006
45 020360
    020360 012637 002244
    020364 012601
    020366 012600
46 020370 000207
47
48 020372 052737 100000 030200
49 020400 000735
50
51 020402 052777 040000 162046
    
```

52	020410	052737	040000	030200	BIS	#BIT14,TDWRD1	:MARK TD MOVED IN TD RECORD
53	020416	026127	000002	140000	CMP	2(R1),#NULL	:ANY MORE DEFECT SKIPS ON THIS TRACK ?
54	020424	001723			BEO	2\$:BR IF NO
55	020426	016146	000002		MOV	2(R1),-(SP)	:SAVE NEXT DEFECT VALUE
56	020432	016046	177776		MOV	-2(R0),-(SP)	:GET UPPER LIMIT FOR TD MOVED
57	020436	161116			SUB	(R1),(SP)	:FIND DIFFERENCE AND
58	020440	162616			SUB	(SP)+,(SP)	:ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
59	020442	016001	000002		MOV	2(R0),R1	:GET UPPER LIMIT FOR HDR 0 MOVED
60	020446	022601			CMP	(SP)+,R1	:WAS HEADER 0 ALSO MOVED ?
61	020450	003311			BGT	2\$:BR IF NO
62	020452	005746			TST	-(SP)	:BACKUP TO GET VALUE AGAIN
63	020454	162701	000066		SUB	#<DS/2>,R1	:SUBTRACT HALF A DEFECT TO GET LOWER LIMIT
64							:FOR HDR 0 MOVED
65	020460	022601			CMP	(SP)+,R1	:WAS HEADER 0 ALSO MOVED ?
66	020462	003704			BLE	2\$:BR IF NO
67	020464	000742			BR	4\$	

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, MSRDTD, TDWRD1, TDBCNT, DESTRK
:
:OUTPUTS ARE: FUNCTN, ERRFNC, ERRMSK, DESTRK, RPOF, BADDR, 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,@RPOF  ;;SET COMMAND MODIFIER BIT
        MOV      #TDWRD1,BADDR ;;GET POINTER TO BEGINNING OF TD RECORD 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
        TST     R1YCNT      ;;DID WE HAVE AN ERROR ?
        BNE    1$         ;;BR IF YES
        ;;READ WAS GOOD, SO NOW COMPARE EXPECTED
        ;;AND RECIEVED DATA
        MOV      #TDWRD1,R0 ;;GET POINTER TO BEGINNING OF TD BUFFER
        MOV      DECYL,TDCPY1 ;;NOW PREPARE TO CHECK THE CORRECTNESS OF THE DATA
        BIS     #BIT13!BIT12,TDCPY1 ;;THIS IS 16 BIT MODE, AND A TD RECORD
        MOV      (R0),TDCPY1 ;;SET REMAINDER OF EXPECTED BITS
        BIS     #377,TDCPY2  ;;SET UP FOR THE SECOND WORD OF THE TD
        MOV      (R0)+,TDCPY1 ;;CYLINDER ADDRESS MATCH ?
        CMP     3$         ;;BR IF NO
        BNE    3$         ;;TRACK/SECTOR ADDRESS MATCH ?
        BEQ     4$         ;;BR IF YES
        MOV      #11,ERRMSK ;;LOG 'TD ADDRESS INCORRECT ERROR' STATUS
        BR      5$         ;;TRY READ AGAIN
        TST     (R0)+     ;;IS IT VALID DATA ?
4$:      TST     (R0)+     ;;IS IT VALID DATA ?
  
```

```

020466 013746 002370
020472 010046
020474 010146
020476 012737 000075 002370
020504 012737 005425 002254
020512 012700 030200
020516 013701 002314
020522 005020
020524 005301
020526 003375
020530 013746 002244
020534 112737 000377 002244
020542 052777 100000 161706
020550 012737 030200 002230
020556 013737 002314 002362
020564 004737 010166
020570 012637 002244
020574 004737 012446
020600 000453
020602 005737 002266
020606 001341
020610 012700 030200
020614 013737 002304 030214
020622 052737 030000 030214
020630 051037 030214
020634 013737 002244 030216
020642 052737 000377 030216
020650 022037 030214
020654 001003
020656 022037 030216
020662 001404
020664 012737 000011 002264
020672 000405
020674 005720
  
```

55	020676	001011				BNE	6\$:BR IF YES
56	020700	012737	000012	002264		MOV	#12,ERRMSK		:LOG 'TD DATA INCORRECT ERROR' STATUS
57	020706	005737	002170		5\$:	TST	SCANR		:ARE WE DOING DISK SCAN ?
58	020712	001006				BNE	7\$:BR IF YES
59	020714	004737	012752			JSR	PC,ERRORS		:REPORT THE ERROR
60	020720	000403				BR	7\$		
61	020722	062766	000002	000006	6\$:	ADD	#2,6(SP)		:ADJUST FOR GOOD RETURN
62	020730				7\$:				
	020730	012601				MOV	(SP)+,R1		::POP STACK INTO R1
	020732	012600				MOV	(SP)+,R0		::POP STACK INTO R0
	020734	012637	002370			MOV	(SP)+,FUNCTN		::POP STACK INTO FUNCTN
63	020740	000207				RTS	PC		:OK, END SUB AND RETURN TO MAIN


```

1      .SBTTL LIST TD(S) MODULE
2
3      :*****
4      :THIS MODULE IS USED TO LIST THE TD(S) THAT HAVE DEFECTS FOUND DURING
5      :THE LIST OPERATION.
6
7      :*
8      :INPUTS ARE:   FROM A BUFFER CALLED 'TDWRD1'
9
10     :*
11     :OUTPUTS ARE:  PRESENTED TO THE USER ON THE CONSOLE PRINTER
12
13     :CALL
14     :
15     :*****
16
17     LISTDS:
18     MOV     FUNCTN,-(SP)      ;;PUSH FUNCTN ON STACK
19     MOV     R1,-(SP)         ;;PUSH R1 ON STACK
20     MOV     R2,-(SP)         ;;PUSH R2 ON STACK
21     MOV     R3,-(SP)         ;;PUSH R3 ON STACK
22     CMP     TDWRD3,#NULL     ;;ARE THERE ANY DEFECTS TO LIST ?
23     BEQ     8$               ;;BR IF NO
24     MOV     #TDDEF,TEXT      ;;LOAD TEXT WITH ADDRESS OF MESSAGE
25     TST     LISHDR           ;;LIST HEADERS IN DEFECT SECTORS ?
26     BEQ     1$               ;;BR IF NO
27     MOV     #ANDHDR,TEXT     ;;LOAD TEXT WITH ADDRESS OF MESSAGE
28     JSR     PC,TDDUMP        ;;GO DUMP THE TRACK DESCRIPTOR
29
30     TST     LISHDR           ;;LIST HEADERS IN DEFECT SECTORS ?
31     BEQ     8$               ;;BR IF NO
32     MOV     #RDHD,FUNCTN     ;;LOAD A READ HEADER COMMAND
33     JSR     PC,MAKTRK        ;;NOW EXECUTE THE COMMAND
34     BR      8$               ;;RETURN HERE IF EXCEEDED RETRY LIMIT
35     ;;ELSE RETURN HERE
36     MOV     BADDR,R1         ;;GET POINTER TO BEGINNING OF HEADER BLOCK
37     MOV     TKWCNT,R3        ;;GET WORD COUNT FOR TRACK
38
39     2$:  CMP     (R1)+,(R1)+   ;;GET OVER HEADER WORDS 1 AND 2
40     CMP     (R1),#NULL       ;;ARE THERE ANY DEFECTS TO LIST IN THIS SECTOR ?
41     BNE     3$               ;;BR IF YES
42     CMP     (R1)+,(R1)+     ;;GET OVER HEADER WORDS 3 AND 4
43     CMP     (R1)+,(R1)+     ;;GET OVER HEADER WORDS 5 AND 6
44     BR      7$               ;;
45     3$:  MOV     -2(R1),-(SP)  ;;GET THE SECOND WORD OF THE TD (TRK/SEC ADRS)
46     MOV     (SP),-(SP)       ;;SAVE SECOND WORD
47     CLRB    3(SP)           ;;GET RID OF HI BYTE (EXTRACT SECTOR)
48     ;;PRINT TD WORD#2, SEC DATA
49
50     MOV     (SP)+,-(SP)      ;;
51     MOV     (SP)+,-(SP)      ;;
52     MOV     #FRMT37,-(SP)   ;;
53     MOV     #3,-(SP)        ;;
54     MOV     SP,R0           ;;
55     TRAP    C$PNTF          ;;
56     ADD     #10,SP          ;;
57     MOV     #4,R2           ;;GET MAXIMUM # OF DEFECTS
58     4$:  CMP     (R1)+,#NULL  ;;IS IT NULL DATA ?
59     BEQ     5$               ;;BR IF YES
60     ;;PRINT THE DEFECT DATA
    
```

48	021132	016146	177776		MOV	-2(R1),-(SP)	
	021136	012746	003624		MOV	#FRMT14, -(SP)	
	021142	012746	000002		MOV	#2, -(SP)	
	021146	010600			MOV	SP, R0	
	021150	104417			TRAP	C\$PNTF	
	021152	062706	000006		ADD	#6, SP	
49	021156	000412			BR	6\$:AND GO-ON
50	021160			5\$:			:PRINT * 140000
51	021160	016146	177776		MOV	-2(R1), -(SP)	
	021164	012746	003635		MOV	#FRMT15, -(SP)	
	021170	012746	000002		MOV	#2, -(SP)	
	021174	010600			MOV	SP, R0	
	021176	104417			TRAP	C\$PNTF	
	021200	062706	000006		ADD	#6, SP	
52	021204	005302		6\$:	DEC	R2	:ONE LESS ITERATION TO-GO
53	021206	003346			BGT	4\$:DO UNTIL = 0!
54							:CR-LF
55	021210	012746	002642		MOV	#CRLF, -(SP)	
	021214	012746	000001		MOV	#1, -(SP)	
	021220	010600			MOV	SP, R0	
	021222	104417			TRAP	C\$PNTF	
	021224	062706	000004		ADD	#4, SP	
56	021230	162703	000006	7\$:	SUB	#6, R3	:DONE ALL SECTORS YET ?
57	021234	003303			BGT	2\$:BR IF NO
58	021236			8\$:			
	021236	012603			MOV	(SP)+, R3	::POP STACK INTO R3
	021240	012602			MOV	(SP)+, R2	::POP STACK INTO R2
	021242	012601			MOV	(SP)+, R1	::POP STACK INTO R1
	021244	012637	002370		MOV	(SP)+, FUNCTN	::POP STACK INTO FUNCTN
59	021250	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 CALLED 'BADDR'
8
9      :OUTPUTS ARE:  PRESENTED TO THE USER ON THE CONSOLE PRINTER
10
11     :CALL
12     :          JSR      PC,NEWLST
13     :*****
14
15     021252  013703  002230  NEWLST:  MOV      BADDR,R3      ;GET THE BUFFER ADDRESS
16     021256  005713          TST      (R3)          ;ANY NEW DEFECTS ?
17     021260  001543          BEQ      5$          ;BR IF NO
18     021262  105237  002236          INCB    SCANIT        ;INCREMENT SCAN ITERATION COUNT
19     021266  005046          CLR      -(SP)        ;SETUP SCAN ITERATION NUMBER FOR TYPEOUT
20     021270  113716  002236          MOVB   SCANIT,(SP)    ;GET THE ITERATION COUNT
21
22
23
24
25
26     021274  012646          MOV      (SP)+,-(SP)  ;
27     021276  012746  004266          MOV      #FRMT32,-(SP)
28     021302  012746  000002          MOV      #2,-(SP)
29     021306  010600          MOV      SP,R0
30     021310  104417          TRAP    C$PNTF
31     021312  062706  000006          ADD     #6,SP
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
    
```

	021426	012646		MOV	(SP)+,-(SP)	
	021430	012646		MOV	(SP)+,-(SP)	
	021432	012746	004530	MOV	#FRMT36,-(SP)	
	021436	012746	000006	MOV	#6,-(SP)	
	021442	010600		MOV	SP,R0	
	021444	104417		TRAP	C\$PNTF	
	021446	062706	000016	ADD	#16,SP	
43	021452	021627	140000	2\$: CMP	(SP),#NULL	:IS IT NULL DATA ?
44	021456	001412		BEQ	3\$:IF MATCH, YES
45						:PRINT THE DATA
46	021460	012646		MOV	(SP)+,-(SP)	
	021462	012746	003624	MOV	#FRMT14,-(SP)	
	021466	012746	000002	MOV	#2,-(SP)	
	021472	010600		MOV	SP,R0	
	021474	104417		TRAP	C\$PNTF	
	021476	062706	000006	ADD	#6,SP	
47	021502	000411		BR	4\$:AND GO-ON
48	021504			3\$:		:PRINT " 140000 "
49	021504	012646		MOV	(SP)+,-(SP)	
	021506	012746	003635	MOV	#FRMT15,-(SP)	
	021512	012746	000002	MOV	#2,-(SP)	
	021516	010600		MOV	SP,R0	
	021520	104417		TRAP	C\$PNTF	
	021522	062706	000006	ADD	#6,SP	
50	021526	021627	177777	4\$: CMP	(SP),#-1	:IS THIS TERMINATOR ?
51	021532	001347		BNE	2\$:BR IF NO
52	021534	005726		TST	(SP)+	:RESTORE STACK
53						:CR-LF
54	021536	012746	005542	MOV	#CRLF,-(SP)	
	021542	012746	005501	MOV	#1,-(SP)	
	021546	010600		MOV	SP,R0	
	021550	104417		TRAP	C\$PNTF	
	021552	062706	005504	ADD	#4,SP	
55	021556	012603		MOV	(SP)+,R3	:GET BACK POINTER TO TD LIST
56	021560	005723		TST	(R3)+	:GET OVER INTERNAL STATUS WORD IN TD LIST
57	021562	020327	117656	CMP	R3,#ENDBUF	:AT END OF BUFFER?
58	021566	101663		BLOS	1\$:BR IF NO
59	021570	000207		5\$: 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

.SBTTL TD SCAN MODULE

```

.....
THIS MODULE IS USED BY THE SCANNER TO DETERMINE IF A TD RECORD IS CORRUPTED.
*
OUTPUTS ARE: TO (R1)+, WHICH IS A BUFFER CALLED 'DEFBUF' AND TO
NEWCNT AND ERRCTL
*
CALL
JSR PC,TDSCAN
.....
    
```

```

TDSCAN:
MOV ERRCTL, -(SP)      ;; PUSH ERRCTL ON STACK
JSR PC,READTD         ;; CALL THE READ TD SUBROUTINE
NOP                   ;; RETURN HERE IF EXCEEDED RETRY LIMIT
                       ;; ELSE RETURN HERE, NO ERROR
1$: SUB ERRCTL, (SP)   ;; GET # OF ERRORS DETECTED DURING READ TD
NEG (SP)              ;; MAKE NUMBER POSITIVE
CMP (SP)+, #2         ;; DID WE GET 2 MORE ERRORS AFTER THE ORIGINAL ?
BGT 2$                ;; BR IF YES
ADD #2, (SP)          ;; MODIFY ADDRESS TO SKIP OVER BRANCH ON RETURN
BR 4$                 ;; AND TAKE RETURN

2$: INC NEWCNT         ;; ADD ONE TO THE TOTAL DEFECT COUNT
MOV DESCYL, (R1)      ;; START FORMING THE BUFFER NOW
BIS #BIT13:BIT12, (R1)+; THIS IS THE FIRST TD WORD.
MOV DESTRK, (R1)      ;; FORM THE SECOND WORD
BIS #377, (R1)+       ;; OF THE TD
BIT #BIT11, 10(R1)   ;; DID WE TRY TO MOVE THE TD ALREADY ?
BEQ 3$                ;; BR IF NO
CLR SUPRSS           ;; ALLOW ERROR MESSAGE TO BE OUTPUT
MOV #1, ERRMSK        ;; LOG 'TD NOT READABLE ERROR' STATUS
CMP -(R1), -(R1)     ;; BACK UP THE POINTERS
BR 4$                 ;; AND TAKE THE RETURN

3$: MOV #<Gi+<DS/2>>, (R1)+ ;; GET THE CONSTANT FOR TD MOVED, WORD #3
MOV #NULL, (R1)+     ;; NULL TD WORD #4,
MOV #NULL, (R1)+     ;; TD WORD #5,
MOV #NULL, (R1)+     ;; AND TD WORD #6.
BIS #BIT1, (R1)+     ;; MARK TD MOVED, IN THE INTERNAL STATUS

4$: RTS PC
    
```

```

021572 013746 002262
021572 004737 020466
021602 000240
021604 163716 002262
021610 005416
021612 022627 000002
021616 003003
021620 062716 000002
021624 000437
021626 005237 002250
021632 013711 002304
021636 052721 030000
021642 013711 002244
021646 052721 000377
021652 032761 000002 000010
021660 001407
021662 005037 002242
021666 012737 000001 002264
021674 024141
021676 000412
021700 012721 000166
021704 012721 140000
021710 012721 140000
021714 012721 140000
021720 052721 000002
021724 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
49
50
51
52
53
54
55
56
57

.SBTTL HEADER SCAN MODULE

 : THIS MODULE IS USED TO DETERMINE IF A HEADER IS CORRUPTED WHEN INVOKED BY THE
 : SCANNER.

 : OUTPUTS ARE: DEFECT IS MOVED TO (R1)+, WHICH IS A BUFFER CALLED 'DEFBUF' OR
 : TO A 2 WORD BUFFER CALLED 'TMPBSF', WHICH IS THEN MOVED TO
 : (R2)+, WHICH IS A BUFFER CALLED 'USRBUF'.

: CALL JSR PC,HDSCAN

```

HDSCAN: MOV #RDHD,FUNCTN ;SETUP FOR READ-HEADER FUNCTION
        MOV #DBUFF,BADDR ;GET POINTER TO BEGINNING OF DATA BUFFER
        MOV #RHDATA,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
        MOV ERRRTL,-(SP) ;:PUSH ERRRTL ON STACK
1$:     BIS #CMOD,@RPOF ;SET COMMAND MODIFIER BIT
        MOV TKWCNT,NEGWRD ;SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
        JSR PC,DRIVER ;DO THE TRANSFER
        JSR PC,RETRY ;SEE IF WE HAD ANY ERRORS
        BR 2$ ;RETURN HERE IF EXCEEDED RETRY LIMIT
        ;ELSE RETURN HERE
        TST RTYCNT ;DID WE HAVE AN ERROR ?
        BNE 1$ ;BR IF YES
2$:     SUB ERRRTL,(SP) ;GET # ERRORS DETECTED DURING READ
        NEG (SP) ;MAKE NUMBER POSITIVE
        CMP (SP)+,#2 ;DID WE GET 2 MORE ERRORS AFTER THE ORIGINAL ?
        BGT 3$ ;BR IF YES
        ADD #2,(SP) ;MODIFY ADDRESS TO SKIP OVER BRANCH ON RETURN
        BR 12$ ;TAKE THE RETURN
3$:     INC NEWCNT ;ADD ONE TO THE TOTAL DEFECT COUNT
        CLR ECCWRD ;USE DATA WORD 0 AS DEFECT TO MOVE HEADER
        JSR PC,FINDSEC ;GO FIND THE SECTOR IN WHICH THE NEW DEFECT
        ;WAS FOUND.
        BEQ 4$ ;BR IF SECTOR 0
        JSR PC,RELTIME ;CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
        BR 6$ ;INSERT DEFECT INTO TD
4$:     BIT #BIT15,TDWRD1 ;IS HDR 0 MOVED ALREADY ?
        BNE 5$ ;BR IF YES
        MOV #<G1+TD+G2+<DS/2>>,ECCWRD ;GET THE CONSTANT FOR HDR 0 MOVED
        BR 6$ ;INSERT DEFECT INTO TD
5$:     BIT #BIT14,TDWRD1 ;IS THE TD MOVED ALREADY ?
        BNE 7$ ;BR IF YES
        MOV #<G1+<DS/2>>,ECCWRD ;GET THE CONSTANT FOR TD MOVED
6$:     JSR PC,INSERT ;INSERT DEFECT INTO TD
52:    TST TMPBSF ;DO WE HAVE ROOM IN THE TD RECORD ?
53:    BMI 8$ ;BR IF YES
54:    MOV SECADD,TMPBSF+2 ;SAVE THE BAD SECTOR AND
55:    JSR PC,LOADBSF ;LOAD THE USR BSF BUFFER
56:    CLR 14(R1) ;RESET THE INTERNAL STATUS
57:    BR 12$
  
```


58										
59	022146	012704	000006		8\$:	MOV	#6,R4	:	GET # OF ENTRIES IN TD RECORD	
60	022152	012703	030200			MOV	#TDWRD1,R3	:	GET POINTER TO BEGINNING OF TD BUFFER	
61	022156	012321			9\$:	MOV	(R3)+,(R1)+	:	PUT NEW DEFECT INTO BUFFER	
62	022160	005304				DEC	R4	:	DONE ALL ENTRIES YET ?	
63	022162	003375				BGT	9\$:	BR IF NO	
64	022164	032737	100000	030200		BIT	#BIT15,TDWRD1	:	IS HDR 0 MOVED ?	
65	022172	001402				BEQ	10\$:	BR IF NO	
66	022174	052711	000004			BIS	#BIT2,(R1)	:	MARK HDR 0 MOVED, IN THE INTERNAL STATUS	
67	022200	032737	040000	030200	10\$:	BIT	#BIT14,TDWRD1	:	IS THE TD MOVED ?	
68	022206	001402				BEQ	11\$:	BR IF NO	
69	022210	052711	000002			BIS	#BIT1,(R1)	:	MARK TD MOVED, IN THE INTERNAL STATUS	
70	022214	005721			11\$:	TST	(R1)+	:	GET OVER THE INTERNAL STATUS WORD	
71	022216	000207			12\$:	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
54
55
56
57

.SBTTL DATA SCAN MODULE

```

:*****
:THIS MODULE IS USED TO FIND ANY DATA FIELD DEFECTS. IT USES A FULL TRACK
:BUFFER FOR DATA TRANSFERS, CALLED 'DBUFF'.
:
:OUTPUTS ARE: DEFECT IS MOVED TO (R1)+, WHICH IS A BUFFER CALLED 'DEFBUF' OR
:              TO A 2 WORD BUFFER CALLED 'TMPBSF', WHICH IS THEN MOVED TO
:              (R2)+, WHICH IS A BUFFER CALLED 'USRBUF'.
:
:CALL
:              JSR      PC,DASCAN
:*****
    
```

```

DASCAN: MOV      #BGNPAT,PATRN      ;START WITH BEGINNING DATA PATTERN
1$:      JSR      PC,GETPAT          ;GET A PATTERN AND SETUP 'PATRN' TO CONTAIN
:              ;THE NEXT DATA PATTERN TO BE USED.
2$:      CLR      R3                 ;CLEAR WRITE CHECK ERROR COUNTER
:              ;GET POINTER TO BEGINNING OF DEFECT BUFFER
:              MOV      #TDSBLK,R4
:              MOV      #WRDAT,FUNCTN ;SETUP FOR A WRITE DATA COMMAND
:              MOV      #WRDAT,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
:              MOV      #DBUFF,BADDR  ;GET POINTER TO BEGINNING OF DATA BUFFER
:              MOV      #<256.*50.>,NEGWRD ;SET FOR A FULL TRACK DATA TRANSFER
:              JSR      PC,DRIVER      ;DO THE TRANSFER
:              JSR      PC,RETRY      ;SEE IF WE HAD ANY ERRORS
:              BR       3$            ;RETURN HERE IF EXCEEDED RETRY LIMIT
:              ;ELSE RETURN HERE
:              TST      RTYCNT        ;DID WE HAVE AN ERROR ?
:              BNE     2$            ;BR IF YES
:              BR       4$            ;GOOD WRITE!!!
3$:      CLR      SUPRSS             ;ALLOW ERROR MESSAGE TO BE OUTPUT
:              MOV      #17,ERRMSK   ;LOG 'UNRECOVERABLE ERROR DURING PACK SCAN' STATUS
:              BR       15$          ;TAKE THE BAD RETURN
4$:      MOV      #WCKD,FUNCTN      ;SET UP FOR A WRITE-CHECK COMMAND
:              MOV      #WCKDAT,ERRFNC ;SAVE FUNCTION FOR ERROR ROUTINE
:              MOV      #<256.*50.>,NEGWRD ;FULL TRACK DATA TRANSFER
:              JSR      PC,DRIVER      ;DO THE WRITE CHECK NOW!
:              JSR      PC,RETRY      ;SEE IF WE HAD ANY ERRORS
:              BR       6$            ;RETURN HERE IF EXCEEDED RETRY LIMIT
:              ;ELSE RETURN HERE
:              TST      RTYCNT        ;DID WE HAVE AN ERROR ?
:              BEQ     7$            ;BR IF NO
:              BIT      #WCE,REG+10  ;IS WCE SET ?
:              BEQ     4$            ;BR IF NO
:              BR       8$            ;GO FIND BAD WORD IN SECTOR
5$:      CLR      SUPRSS             ;ALLOW ERROR MESSAGE TO BE OUTPUT
:              MOV      #17,ERRMSK   ;LOG 'UNRECOVERABLE ERROR DURING PACK SCAN' STATUS
:              BR       15$          ;TAKE THE BAD RETURN
6$:      CLR      SUPRSS             ;ALLOW ERROR MESSAGE TO BE OUTPUT
:              MOV      #17,ERRMSK   ;LOG 'UNRECOVERABLE ERROR DURING PACK SCAN' STATUS
:              BR       15$          ;TAKE THE BAD RETURN
7$:      CMP      PATRN,#ENDPAT      ;DONE WITH ALL PATTERNS YET ?
:              BLO     1$            ;BR IF NO
:              ADD     #2,(SP)       ;RETURN WITH NO ERROR (THE TRACK OK)
:              BR       15$          ;TAKE THE GOOD RETURN
    
```


DATA SCAN MODULE

58	022434	005037	002266		8\$:	CLR	RTYCNT		:RESET RETRY COUNT TO 0
59	022440	004737	022634			JSR	PC,FINDWRD		:GO FIND THE BAD WORD WITHIN THE SECTOR OF THE
60									:NEW DEFECT.
61	022444	004737	022712			JSR	PC,FINDSEC		:GO FIND THE SECTOR IN WHICH THE NEW DEFECT
62									:WAS FOUND.
63	022450	004737	014372			JSR	PC,RELTIME		:CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
64	022454	013714	002364			MOV	ECCWRD,(R4)		:LOG THE NEW DEFECT IN DEFECT BUFFER
65	022460	004737	022744			JSR	PC,TOLER		:SEE IF WCE'S ARE WITHIN TOLERANCE AND UPDATE
66									:POINTER TO DEFECT TABLE IN R4.
67	022464	022737	000020	002264		CMP	#20,ERRMSK		:DID RANDOM WCE'S OCCUR ?
68	022472	001457				BEQ	15\$:BR IF YES
69	022474	005203				INC	R3		
70	022476	020327	000002			CMP	R3,#2		:DID WE GET 2 MORE ERRORS AFTER THE ORIGINAL ?
71	022502	003711				BLE	4\$:BR IF NO
72	022504	023737	002364	002340		CMP	ECCWRD,TDLMTS+2		:WILL BEGINNING OF DEFECT BE IN GAP 2 ?
73	022512	003005				BGT	9\$:BR IF NO
74	022514	013737	002340	002364		MOV	TDLMTS+2,ECCWRD		:GET UPPER LIMIT FOR HDR 0 MOVED AND
75	022522	005237	002364			INC	ECCWRD		:ADD ONE TO IT.
76	022526	005237	002250		9\$:	INC	NEWCNT		:ADD ONE TO THE TOTAL DEFECT COUNT
77	022532	004737	014522			JSR	PC,INSERT		:INSERT DEFECT INTO TD
78									
79	022536	005737	030244		10\$:	TST	TMPBSF		:DO WE HAVE ROOM IN THE TD RECORD ?
80	022542	100406				BMI	11\$:BR IF YES
81	022544	113737	002366	030246		MOV	SECADD,TMPBSF+2		:SAVE THE BAD SECTOR AND
82	022552	004737	011122			JSR	PC,LODBSF		:LOAD THE USR BSF BUFFER
83	022556	000425				BR	15\$:TAKE THE RETURN
84									
85	022560	012704	000006		11\$:	MOV	#6,R4		:GET # OF ENTRIES IN TD RECORD
86	022564	012703	030200			MOV	#TDWRD1,R3		:GET POINTER TO BEGINNING OF TD BUFFER
87	022570	012321			12\$:	MOV	(R3)+,(R1)+		:PUT THE NEW DEFECT INTO BUFFER
88	022572	005304				DEC	R4		:DONE ALL ENTRIES YET ?
89	022574	003375				BGT	12\$:BR IF NO
90	022576	032737	100000	030200		BIT	#BIT15,TDWRD1		:IS HDR 0 MOVED ?
91	022604	001402				BEQ	13\$:BR IF NO
92	022606	052711	000004			BIS	#BIT2,(R1)		:MARK HDR 0 MOVED, IN THE INTERNAL STATUS
93	022612	032737	040000	030200	13\$:	BIT	#BIT14,TDWRD1		:IS THE TD MOVED ?
94	022620	001402				BEQ	14\$:BR IF NO
95	022622	052711	000002			BIS	#BIT1,(R1)		:MARK TD MOVED, IN THE INTERNAL STATUS
96	022626	052721	000200		14\$:	BIS	#BIT7,(R1)+		:MARK DATA AREA DEFECT, IN THE INTERNAL STATUS
97	022632	000207			15\$:	RTS	PC		:TAKE 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
47
48
49
50
51

```

:*****
: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: ECCWRD (WORD COUNT IN SECTOR)
:
:CALL
:      JSR      PC,FINDWRD      ;CALL THE ROUTINE
:*****
    
```

```

15 022634 013746 002502 FINDWRD: MOV REG+2,-(SP) ;PUT RPWC CONTENTS ON THE STACK
16 022640 062716 031000 ADD #<256.*50.>,(SP) ;AND CALCULATE THE # OF WORDS XFERD
17 022644 005316 DEC (SP) ;SUBTRACT 1 FOR AN ODD OR EVEN WORD
18 022646 005737 002416 TST RHTYPE ;WAS THE CONTROLLER AN RH70 ?
19 022652 001407 BEQ 1$ ;BR IF 0, NO
20 022654 162716 000002 SUB #2,(SP) ;SUBTRACT 2 FOR A DOUBLE WORD
21 022660 032737 004000 002552 BIT #BIT11,REG+52 ;WAS WCE ON AN EVEN WORD?
22 022666 001401 BEQ 1$ ;BR IF 0, NO
23 022670 005316 DEC (SP) ;SUBTRACT 1 FOR AN EVEN WORD
24 022672 162716 000400 1$: SUB #256.,(SP) ;DID WE FIND THE BAD SECTOR YET ?
25 022676 002375 BGE 1$ ;BR IF NO
26 022700 062716 000400 ADD #256.,(SP) ;THIS IS THE BAD WORD IN THE BAD SECTOR
27 022704 012637 002364 MOV (SP)+,ECCWRD ;SAVE WORD COUNT IN SECTOR
28 022710 000207 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
:*****
    
```

```

44 022712 005037 002366 FINDSEC: CLR SECADD ;INIT SECTOR ADDRESS COUNT
45 022716 113737 002506 002366 MOVB REG+6,SECADD ;LOAD SECTOR ADDRESS IN ERROR
46 022724 001004 BNE 1$ ;BR IF ERROR NOT IN LAST SECTOR, ELSE
47 022726 113737 002306 002366 MOVB SEC50,SECADD ;SETUP FOR ERROR IN LAST SECTOR
48 022734 000402 BR 2$
49
50 022736 005337 002366 1$: DEC SECADD ;ADJUST RPDA TO ACCOUNT FOR ERROR
51 022742 000207 2$: RTS PC ;AND TAKE THE RETURN
    
```



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16 022744 012700 023050
17 022750 022704 033400
18 022754 001003
19 022756 011410
20 022760 011460 000002
21 022764 021410
22 022766 002402
23 022770 011410
24 022772 000405
25 022774 005720
26 022776 021410
27 023000 003002
28 023002 011410
29 023004 005740
30 023006 012046
31 023010 011646
32 023012 061016
33 023014 006216
34 023016 012637 002364
35 023022 161016
36 023024 022627 000006
37 023030 003405
38 023032 005037 002242
39 023036 012737 000020 002264
40 023044 005724
41 023046 000207
42
43 023050 000000
44 023052 000000

:*****
: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
:TD RECORD AND THE USER IS NOTIFIED OF THE PROBLEM.
:*
:INPUTS ARE: FROM A BUFFER CALLED 'TDSBLK'
:*
:OUTPUTS ARE: ECCWRD (AVERAGE DEFECT), ERRMSK (IF ERROR DETECTED)
:*
:CALL
: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$: MOV (R0)+,-(SP) ;GET HI VALUE AND
MOV (SP),-(SP) ;SAVE IT.
ADD (R0),(SP) ;TOTAL THE HI AND LO VALUES AND
ASR (SP) ;DETERMINE THE AVERAGE.
MOV (SP)+,ECCWRD ;SAVE AVERAGE VALUE AS DEFECTIVE SPOT
SUB (R0),(SP) ;GET DIFFERENCE FROM LO TO HI VALUES
CMP (SP)+,#6 ;ARE HI/LO VALUES WITHIN +/-3 WORDS ?
BLE 4$ ;BR IF YES
CLR SUPRSS ;ALLOW ERROR MESSAGE TO BE OUTPUT
MOV #20,ERRMSK ;LOG 'RANDOM WCE 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

```

1
2
3
4
5
6
7
8
9
10
11
12
13

```

*****
: THIS MODULE PROVIDES TEST PATTERNS TO BE USED DURING THE SCANNER PROCESS.
:
: INPUTS ARE: FROM BGNPAT - ENDPAT (BEGIN PATTERN THRU END PATTERN)
:
: OUTPUTS ARE: TO A BUFFER CALLED 'DBUFF'
:
: CALL
: JSR PC,GETPAT
*****
    
```

```

13 023054
    023054 010146
    023056 010246
    023060 010346
14 023062 013703 002312
15 023066 012702 031000
16 023072 012701 035660
17 023076 012321
18 023100 011321
19 023102 005743
20 023104 162702 000002
21 023110 003372
22 023112 022323
23 023114 010337 002312
24 023120 012603
    023122 012602
    023124 012601
25 023126 000207
    
```

```

GETPAT:
MOV R1,-(SP)      ;;PUSH R1 ON STACK
MOV R2,-(SP)      ;;PUSH R2 ON STACK
MOV R3,-(SP)      ;;PUSH R3 ON STACK
MOV PATRN,R3      ;;GET THE NEW PATTERN
MOV #<256.*50.>,R2 ;;GET A WORD/TRACK COUNT
MOV #DBUFF,R1     ;;GET THE READ/WRITE BUFFER
1$: MOV (R3)+,(R1)+ ;;WRITE FIRST WORD OF PATTERN
    MOV (R3),(R1)+ ;;NOW WRITE THE SECOND WORD
    TST -(R3)      ;;BACK UP THE INPUT POINTER
    SUB #2,R2      ;;REDUCE THE ITERATION COUNT
    BGT 1$         ;;IF > 0, KEEP GOING
    CMP (R3)+,(R3)+ ;;UPDATE TO NEXT 2 WORD PATTERN
    MOV R3,PATRN   ;;STORE THE NEW PATTERN
    MOV (SP)+,R3   ;;POP STACK INTO R3
    MOV (SP)+,R2   ;;POP STACK INTO R2
    MOV (SP)+,R1   ;;POP STACK INTO R1
    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
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50

```

023130
023130 010146
023132 010246
14 023134 012701 030204
15 023140 012702 000004
16 023144 062137 002364
17 023150 005302
18 023152 003374
19 023154 012701 000155
20 023160 062701 000517
21 023164 005037 002366
22 023170 020137 002364
23 023174 103006
24 023176 005237 002366
25 023202 162737 000517 002364
26 023210 100372
27 023212
023212 012602
023214 012601
28 023216 000207
023220
023220 010146
023222 010346
43 023224 004737 020466
44 023230 000401
45 023232 000402
46 023234 004737 025112
47 023240
023240 013746 002364
48 023244 004737 014522
49 023250 012637 002364
50 023254 005737 030244
    
```

```

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

```

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$: ADD (R1)+,ECCWRD ;ADD THE LIST OF DEFECT TO THE NEW DEFECT
DEC R2 ;ONE LESS RECORD ENTRY
BGT 1$ ;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,ECCWRD ;NOW SCALE THE VALUES
BHS 3$ ;IF R1 IS GREATER, SECTOR 0 IS BAD
2$: INC SECADD ;NEXT SECTOR ADDRESS
SUB #<HDR+G3+DATA+G4>,ECCWRD ;REDUCE THE REMAINING WORD COUNT
BPL 2$ ;OF <0, WE'VE FOUND THE BAD SECTOR
3$:
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
RTS PC ;RETURN TO CALLER
    
```

```

*****
: THIS MODULE OUTPUTS DATA TO THE NEW DEFECT BUFFER FOR EVENTUAL TD REWRITE OR
: IT PRODUCES OUTPUT TO THE BAD SECTGR FILE WHEN A TD IS FOUND TO CONTAIN 4
: DEFECTS.
:
: OUTPUTS ARE: TO (R2)+, A BUFFER CALLED 'USRBUF' 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 ECCWRD,-(SP) ;;PUSH ECCWRD ON STACK
JSR PC,INSERT ;INSERT THE DEFECT IN TD
MOV (SP)+,ECCWRD ;;POP STACK INTO ECCWRD
TST TMPBSF ;DO WE HAVE ROOM IN THE TD RECORD ?
    
```

DATA SCAN MODULE

```

51 023260 100410          BMI      3$          ;BR IF YES
52 023262 004737 023130  JSR      PC,FINDBAD  ;FIND THE BAD SECTOR
53 023266 113737 002366  MOVB    SECADD,TMPBSF+2 ;SAVE THE BAD SECTOR AND
54 023274 004737 011122  JSR      PC,LOADBSF   ;LOAD THE USR BSF BUFFER
55 023300 000413          BR       5$
56
57 023302 005237 002250  3$:     INC      NEWCNT   ;ADD ONE TO THE TOTAL DEFECT COUNT
58 023306 012703 000006  MOV     #6,R3        ;GET # OF ENTRIES IN TD RECORD
59 023312 012701 030200  MOV     #TDWRD1,R1   ;GET POINTER TO BEGINNING OF TD BUFFER
60 023316 012124 4$:     MOV     (R1)+,(R4)+ ;PUT THE NEW DEFECT INTO BUFFER
61 023320 005303          DEC      R3          ;DONE ALL ENTRIES YET ?
62 023322 003375          BGT     4$          ;BR IF NO
63 023324 052724 000400  BIS     #BIT8,(R4)+ ;MARK A SCRATCH, IN THE INTERNAL STATUS
64 023330 5$:
65 023330 012603          MOV     (SP)+,R3    ;:POP STACK INTO R3
66 023332 012601          MOV     (SP)+,R1    ;:POP STACK INTO R1
67 023334 000207          RTS      PC         ;NOW RETURN

```

```

:*****
:THIS MODULE DETERMINES IF DATA FOUND IN THE NEW DEFECT BUFFER IS WITHIN
:TOLERANCE TO BE CONSIDERED AS A POSSIBLE SCRATCH. TOLERANCE IS +/- 3 WORDS.
:*****

```

```

72 023336          TOLRAN:
73 023336 010146          MOV     R1,-(SP)    ;:PUSH R1 ON STACK
74 023340 013746 002352  MOV     TEMP1,-(SP) ;:PUSH TEMP1 ON STACK
75 023344 013746 002354  MOV     TEMP2,-(SP) ;:PUSH TEMP2 ON STACK
76 023350 012701 000003  MOV     #3,R1      ;THIS IS THE TOLERANCE VALUE
77 023354 023737 002352  1$:     CMP     TEMP1,TEMP2 ;DO THE INPUTS MATCH?
78 023362 001415          BEQ     4$          ;IF EQUAL YES -RESTORE STACK AND RETURN
79 023364 103403          BLO     2$          ;THEY DO NOT, TEMP 1< TEMP2 (BRANCH)
80 023366 005237 002354  INC     TEMP2      ;ADD ONE TO TEMP2 (TOO SMALL)
81 023372 000402          BR      3$          ;AND REDUCE THE ITERATIONS LEFT
82 023374 005337 002354  2$:     DEC     TEMP2   ;SUBTRACT ONE FROM TEMP 2 (TOO LARGE)
83 023400 005301 3$:     DEC     R1         ;ONE LESS ITERATION TO GO
84 023402 003364          BGT     1$          ;OF >0 DO AGAIN
85 023404 012637 002354  MOV     (SP)+,TEMP2 ;:POP STACK INTO TEMP2
86 023410 012637 002352  MOV     (SP)+,TEMP1 ;:POP STACK INTO TEMP1
87 023414 000401          BR      5$          ;AND RETURN
88 023416 022626 4$:     CMP     (SP)+,(SP)+ ;POP STACK TWICE
89 023420 5$:
90 023420 012601          MOV     (SP)+,R1   ;:POP STACK INTO R1
91 023422 000207          RTS      PC         ;NOW RETURN

```

```

:*****
:THIS MODULE LOOKS AT A 4 WORD FIELD IN THE NEW DEFECT BUFFER, CALLS 'TOLRAN'
:AND DETERMINES IF A SCRATCH DOES INDEED EXIS".
:*****

```

```

93 023424          CHEKTD:
94 023424 010146          MOV     R1,-(SP)    ;:PUSH R1 ON STACK
95 023426 010246          MOV     R2,-(SP)    ;:PUSH R2 ON STACK
96 023430 062701 000004  ADD     #4,R1       ;MOVE R1 UP TO THE FIRST DEFECT
97 023434 005037 002354  CLR     TEMP2      ;TEMP2 IS THE DEFECT DATA ACCUMULATOR
98 023440 012702 000004  MOV     #4,R2      ;ALLOW 4 DEFECTS MAX!
99 023444 062137 002354  1$:     ADD     (R1)+,TEMP2 ;ACCUMULATE A DEFECT
100 023450 004737 023336 JSR     PC,TOLRAN   ;CHECK ITS TOLERANCE

```


99	023454	005737	002354	TST	TEMP2	:IF IT'S OK, TEMP2 <>0!
100	023460	001002		BNE	2\$:IF OK, TAKE BRANCH
101	023462	005302		DEC	R2	:ONE LESS DATA ITEM
102	023464	003367		BGT	1\$:IF >0, KEEP GOING
103	023466		2\$:			
	023466	012602		MOV	(SP)+,R2	::POP STACK INTO R2
	023470	012601		MOV	(SP)+,R1	::POP STACK INTO R1
104	023472	000207		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
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

023474
 023474 010146
 023476 062701 000016
 023502 126137 000003 002245
 023510 001007
 023512 004737 023424
 023516 005737 002354
 023522 001402
 023524 005237 002356
 023530 062701 000016
 023534 020127 035656
 023540 101760
 023542 012601
 023544 000207
 023546
 023546 010146
 023550 010246
 023552
 023552 011146
 023554 042716 170000
 023560 022637 002356
 023564 001406
 023566 062701 000016
 023572 020127 035656
 023576 101765
 023600 000453
 023602 126137 000003 002245
 023610 001406
 023612 062701 000016
 023616 020127 035656
 023622 101767
 023624 101041

```

.....
: 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$:   ADD   #<7*2>,R1        ;; NEXT ENTRY
      CMP   R1,#ENDDEF      ;; OUT OF DEFECT BUFFER YET ?
      BLOS  1$              ;; BR IF NO
      MOV   (SP)+,R1        ;; POP STACK INTO R1
      RTS   PC              ;; RETURN
.....

```

```

.....
: THIS MODULE LOOKS FOR GAPS IN THE SCRATCH DATA CONTAINED WITHIN THE NEW
: DEFECT BUFFER.
.....

```

```

: OUTPUTS ARE:  'HICYL' (IF A GAP IS FOUND WITHIN 5 CYLINDERS OF THE VALUE
:                SET BY 'HIVALU' OUTPUT)
.....

```

```

: CALL
:   JSR   PC,GAPVAL
.....

```

```

GAPVAL:
      MOV   R1,-(SP)          ;; PUSH R1 ON STACK
      MOV   R2,-(SP)          ;; PUSH R2 ON STACK
1$:   MOV   (R1),-(SP)        ;; PUSH (R1) ON STACK
      BIC   #170000,(SP)     ;; STRIP THE UNWANTED BITS
      CMP   (SP)+,HICYL      ;; MATCH?
      BEQ   2$              ;; IF EQUAL, THEY MATCH
      ADD   #<7*2>,R1        ;; GET NEXT BUFFER ENTRY
      CMP   R1,#ENDDEF      ;; OUT OF DEFECT BUFFER YET ?
      BLOS  1$              ;; BR IF NO
      BR    6$              ;; DONE, IF OVERRUN
2$:   CMPB  3(R1),DESTRK+1   ;; IS THIS THE CORRECT TRACK ADDRESS?.
      BEQ   3$              ;; IF =, YES
      ADD   #<7*2>,R1        ;; GET NEXT BUFFER ENTRY
      CMP   R1,#ENDDEF      ;; OUT OF DEFECT BUFFER YET ?
      BLOS  2$              ;; BR IF NO
      BHI   6$              ;; DONE IF OVERRUN
.....

```



```

54 023626 012702 000005      3$:  MOV      #5,R2          :GET THE DELTA VALUE
55 023632 011137 002304      MOV      (R1),DESCYL     :SAVE THE CURRENT CYLINDER ADDRESS
56 023636 005237 002304      INC      DESCYL          :AND LOOK FOR THE NEXT VALUE
57 023642 062701 000016      4$:  ADD      #<7*2>,R1     :GET NEXT BUFFER VALUE
58 023646 020127 035656      CMP      R1,#ENDDEF      :OUT OF DEFECT BUFFER YET ?
59 023652 101024                BHI      6$              :BR IF YES
60 023654 021137 002304      CMP      (R1),DESCYL     :CYLINDER ADDRESS MATCH?
61 023660 103770                BLO      4$              :TAKE BRANCH IF NOT
62 023662 126137 000003 002245  CMPB     3(R1),DESTRK+1   :TRACK ADDRESS MATCH?
63 023670 001405                BEQ      5$              :IF SO TAKE BRANCH
64 023672 005237 002304      INC      DESCYL          :NEXT CYL ADDRESS-PLEASE
65 023676 005302                DEC      R2              :ONE LESS ITERATION TO GO
66 023700 003360                BGT      4$              :IF >0, WE'RE NOT DONE
67 023702 000412                BR       6$              :IF 0 WE NEVER FOUND A CONTINUATION
68 023704 004737 023424      5$:  JSR      PC,CHEKTD       :GET THE NEW TD SUMMATION
69 023710 005737 002354      TST      TEMP2           :IF A SCRATCH VALUE FOUND, THIS <>0
70 023714 001405                BEQ      6$              :TAKE BRANCH IF SCRATCH VALUE NOT FOUND
71 023716 012746 000005      MOV      #5,-(SP)        :GET THE MAX VALUE FOR INTERPOLATION
72 023722 160216                SUB      R2,(SP)         :AND DERIVE THE IMPLIED SCRATCH WIDTH
73 023724 062637 002356      ADD      (SP)+,HICYL     :ADD THAT WIDTH TO THE HIGH LIMIT
74 023730                6$:  MOV      (SP)+,R2        ::POP STACK INTO R2
      023730 012602                MOV      (SP)+,R1        ::POP STACK INTO R1
      023732 012601                RTS      PC              :RETURN TO CALLER
75 023734 000207
76
77
78 .....
79 :THIS MODULE IS USED TO DETERMINE THE HIGHEST VALUE OF A SCRATCH FOUND WITHIN
80 :THE NEW DEFECT BUFFER.
81 :
82 :OUTPUTS ARE:  HICYL
83 :
84 :CALL
85 :      JSR      PC,MAXVAL
86 :.....
87 MAXVAL:
88 023736 010146 000016      MOV      R1,-(SP)        ::PUSH R1 ON STACK
89 023740 062701 000016      ADD      #<7*2>,R1       :GET THE NEXT VALUE PAST THE GAP VALUE
90 023744 011146                1$:  MOV      (R1),-(SP)     ::PUSH (R1) ON STACK
91 023746 042716 170000      BIC      #170000,(SP)    :STRIP THE UNWANTED BITS
92 023752 022637 002356      CMP      (SP)+,HICYL     :MATCH?
93 023756 001406                BEQ      3$              :IF SO, TAKE BRANCH
94 023760 062701 000016      2$:  ADD      #<7*2>,R1     :GET NEXT BUFFER ITEM
95 023764 020127 035656      CMP      R1,#ENDDEF      :OUT OF DEFECT BUFFER YET ?
96 023770 101765                BLOS    1$              :BR IF NO
97 023772 000414                BR       4$
98 023774 126137 000003 002245  3$:  CMPB     3(R1),DESTRK+1   :DO WE HAVE A TRACK ADDRESS MATCH?
99 024002 001366                BNE      2$              :IF NOT, TAKE THE BRANCH
100 024004 004737 023424      JSR      PC,CHEKTD       :AND CHECK THE EXISTING TD RECORD VALUES
101 024010 005737 002354      TST      TEMP2           :CHECK TEMP2 TO DETECT A MATCH
102 024014 001761                BEQ      2$              :IF ZERO, NO MATCH
103 024016 005237 002356      INC      HICYL           :INCREASE THE SCRATCH'S HIGHEST CYLINDER
104 024022 000756                BR       2$              :AND LOOP
105 024024 062737 000005 002356  4$:  ADD      #5,HICYL        :ADD ANOTHER 5 TO THE SCRATCH'S HIGHEST CYLINDER
106 024032 023737 002356 002206      CMP      HICYL,MAXCYL    :LEGAL ADDRESS?
    
```

```

DATA SCAN MODULE

107 024040 101403
108 024042 013737 002206 002356
109 024050 012601
110 024052 000207
111
112
113
114
115
116
117
118
119 024054 010146
    024054 010246
120 024060 010102
121 024062 062702 000016
122 024066 012221
123 024070 020227 035656
124 024074 101774
125 024076 012602
    024100 012601
126 024102 000207
127
128
129
130
131
132
133
134
135 024104 013746 002304
    024110 013746 002244
136 024114 012737 001167 002360
137 024122 012737 177777 002356
138 024130 012704 035660
139 024134 005003
140 024136 012701 032250
141 024142 012105
142 024144 042705 170000
143 024150 011146
144 024152 105016
145 024154 000316
146 024156 022603
147 024160 001415
148 024162 062701 000014
149 024166 005711
150 024170 001403
151 024172 020127 035656
152 024176 101761
153 024200 005203
154 024202 020327 000037
155 024206 101753
156 024210 000137 024720
157

      BLOS 5$          :IF LOWER OR SAME, YES
      MOV  MAXCYL,WICYL :SET UP FOR ONLY THE MAX CYLINDER
5$:
      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.
:*****

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

:*****
:THIS MODULE IDENTIFIES DATA WHICH REPRESENTS A SCRATCH ON THE MEDIA. THE
:DATA FOUND MUST INITIALLY BE CONTIGUOUS, ON THE SAME TRACK - ADJACENT
:CYLINDERS. GAPS AND THE MAXIMUM AND MINIMUM INTERPOLATED ADDRESSES ARE THEN
:DERIVED AND LOGGED IN THE NEW DEFECT BUFFER FOR EVENTUAL TO REWRITE.
:*****

FINSCR:
      MOV  DESCYL,-(SP) :PUSH DESCYL ON STACK
      MOV  DESTRK,-(SP) :PUSH DESTRK ON STACK
      MOV  #631,LOCYL   :SET LO CYL BOUNDARY AND THE
      MOV  #-1,WICYL    :AND THE HI CYL BOUNDARY TO ILLEGAL LIMITS
      MOV  #DBUFF,R4    :GET POINTER TO BEGINNING OF DATA BUFFER
      CLR  R3           :R3 USED TO DETECT A TRACK MATCH
1$:
      MOV  #DEFBUF,R1   :GET POINTER TO BEGINNING OF DEFECT BUFFER
2$:
      MOV  (R1)+,R5     :SAVE FIRST WORD (CYL ADRS)
      BIC  #170000,R5   :STRIP THE STATUS BITS
      MOV  (R1),-(SP)   :SAVE SECOND WORD (TRK/SEC ADRS)
      CLRB (SP)        :GET RID OF LOW BYTE (EXTRACT TRACK) AND
      SWAB (SP)        :MAKE IT THE LOW BYTE OF DATA.
      CMP  (SP)+,R3     :IS THERE A DEFECT ON THIS TRACK ?
      BEQ  5$           :BR IF YES
3$:
      ADD  #<6*2>,R1    :POINT TO NEXT DEFECT IN TABLE
      TST  (R1)         :IS IT VALID DATA ?
      BEQ  4$           :BR IF NO
      CMP  R1,#ENDDEF   :END OF DEFECT BUFFER YET ?
      BLOS 2$           :BR IF NO
4$:
      INC  R3           :GET NEXT TRACK
      CMP  R3,#31.     :LAST TRACK YET ?
      BLOS 1$           :BR IF NO
      JMP  21$         :NO DEFECTS FOUND
    
```


DATA SCAN MODULE

158	024214	010137	002214	5\$:	MOV	R1,TEMPA	:SAVE THE CURRENT TRACK	
159	024220	005205			INC	R5	:GET NEXT CYLINDER	
160	024222	005761	000016	6\$:	TST	16(R1)	:IS IT VALID DATA ?	
161	024226	001764			BEQ	4\$:BR IF NO	
162	024230	016646	000016		MOV	16(SP),-(SP)	:SAVE SECOND WORD (TRK/SEC ADRS)	
163	024234	105016			CLRB	(SP)	:GET RID OF LOW BYTE (EXTRACT TRACK) AND	
164	024236	000316			SWAB	(SP)	:MAKE IT THE LOW BYTE OF DATA.	
165	024240	022603			CMP	(SP)+,R3	:IS THERE A DEFECT ON THIS TRACK ?	
166	024242	001410			BEQ	7\$:BR IF YES	
167	024244	062701	000016		ADD	#<7*2>,R1	:GET THE NEXT ENTRY, PLEASE	
168	024250	005711			TST	(R1)	:IS IT VALID DATA ?	
169	024252	001752			BEQ	4\$:BR IF NO	
170	024254	020127	035656		CMP	R1,#ENDDEF	:END OF DEFECT BUFFER YET ?	
171	024260	101760			BLOS	6\$:BR IF NO	
172	024262	101346			BHI	4\$:BR IF YES	
173	024264	016146	000014	7\$:	MOV	14(R1),-(SP)	:SAVE FIRST WORD (CYL ADRS)	
174	024270	042716	170000		BIC	#170000,(SP)	:STRIP THE STATUS BITS	
175	024274	022605			CMP	(SP)+,R5	:IS THERE A DEFECT ON THIS CYLINDER ?	
176	024276	001331			BNE	3\$:BR IF NO	
177	024300	062701	000016		ADD	#<7*2>,R1	:GET THE LAST LOOKED AT ENTRY	
178	024304	005037	002352		CLR	TEMP1	:THESE ARE USED AS DATA	
179	024310	005037	002354		CLR	TEMP2	:DEFECT ACCUMULATORS	
180	024314	012746	000004		MOV	#4,-(SP)	:GET # OF DEFECT ENTRIES IN TD RECORD	
181	024320	013705	002214		MOV	TEMPA,R5	:GET THE ORIGINAL BUFFER ADDRESS OF THE SCRATCH	
182	024324	005725			TST	(R5)+	:MOVE UP TO THE FIRST DEFECT VALUE	
183	024326	005721			TST	(R1)+	:NEXT DEFECT, PLEASE	
184	024330	062537	002352	8\$:	ADD	(R5)+,TEMP1	:ACCUMULATE THIS INFORMATION	
185	024334	062137	002354		ADD	(R1)+,TEMP2	:FOR BOTH CYLINDERS WITH A COMMON TRACK	
186	024340	004737	023336		JSR	PC,TOLRAN	:IS THE DATA WITHIN TOLERANCE?	
187	024344	023737	002352	002354	CMP	TEMP1,TEMP2	:IF SO, THESE WILL MATCH...	
188	024352	001404			BEQ	9\$:TAKE BRANCH IF SO	
189	024354	005316			DEC	(SP)	:NEXT RECORD ENTRY, PLEASE	
190	024356	003364			BGT	8\$:TRY AGAIN, IF NOT 0	
191	024360	005726			TST	(SP)+	:RESTORE THE STACK	
192	024362	000677			BR	3\$:NO MATCH, KEEP LOOKING	
193	024364	024145		9\$:	CMP	-(R1),-(R5)	:BACK UP THE POINTERS TO THE LAST DEFECT	
194	024366	021627	000004		CMP	(SP),#4	:WHICH DEFECT MATCHED?	
195	024372	001402			BEQ	10\$:IF (SP) MATCHES 4, IT IS THE CURRENT ONE.	
196	024374	005216			INC	(SP)	:AND TALLY THE EVENT	
197	024376	000772			BR	9\$:KEEP GOING	
198	024400	005726		10\$:	TST	(SP)+	:RESTORE THE STACK	
199	024402	016537	177774	002360	MOV	-4(R5),LOCYL	:GET THE LO CYLINDER ADDRESS FOR THE SCRATCH	
200	024410	042737	170000	002360	BIC	#170000,LOCYL	:STRIP THE MASK BITS	
201	024416	023727	002360	000005	CMP	LOCYL,#5	:IS THE CYLINDER ADDRESS > 5?	
202	024424	101004			BHI	11\$:IF SO, TAKE BRANCH	
203	024426	013737	002204	002360	MOV	MINCYL,LOCYL	:SET UP FOR THE FIRST LEGAL CYLINDER	
204	024434	000412			BR	12\$:AND KEEP GOING	
205	024436	162737	000005	002360	11\$:	SUB	#5,LOCYL	:REDUCE THE LOWEST SCRATCH ADDRESS BY 5
206	024444	023737	002360	002204	CMP	LOCYL,MINCYL	:LEGAL ADDRESS?	
207	024452	103005			BHIS	12\$:IF =>, YES	
208	024454	013737	002204	002360	MOV	MINCYL,LOCYL	:CREATE THE LEGAL ADDRESS	
209	024462	013737	002360	002304	12\$:	MOV	LOCYL,DESCYL	:SET UP FOR THE DRIVER
210	024470	162701	000004		SUB	#4,R1	:MOVE R1 TO CYL ADDRESS OF BEGINNING OF SCRATCH	
211	024474	011137	002356		MOV	(R1),MICYL	:SET UP THE FOUND HIGHEST CYL ADDRESS OF THE SCRATCH	
212	024500	005037	002244		CLR	DESTRK	:DESTRK = 0.	
213	024504	116137	000003	002245	MOVB	3(R1),DESTRK+1	:SET UP FOR THE DRIVER	
214	024512	042737	170000	002356	BIC	#170000,MICYL	:STRIP THE UNWANTED MASK	

215	024520	004737	023474		JSR	PC,HIVALU	:GET THE HIGHEST FOUND VALUE	
216	024524	004737	023546		JSR	PC,GAPVAL	:LOOK FOR GAPS; IE HEAD BOUNCES	
217	024530	004737	023736		JSR	PC,MAXVAL	:GET THE LARGEST SCRATCH VALUE	
218	024534	024545			CMP	-(R5),-(R5)	:BACK UP THE POINTER TO BEGINNING OF THE DEFECT RECORD	
219	024536	010501			MOV	R5,R1	:GET THE ORIGINAL SCRATCH DATA	
220	024540	013737	002352	002364	MOV	TEMP1,ECCWRD	:LOAD THE ECC WORD DATA	
221	024546	013737	002360	002304	MOV	LOCYL,DESCYL	:GET THE STARTING CYLINDER ADDRESS	
222	024554			13\$:				
	024554	011146			MOV	(R1),-(SP)	::PUSH (R1) ON STACK	
223	024556	042716	170000		BIC	#170000,(SP)	:STRIP THE UNWANTED MASK	
224	024562	022637	002360		CMP	(SP)+,LOCYL	:IN RANGE?	
225	024566	001416			BEQ	17\$:TAKE BRANCH IF SO	
226	024570	004737	023220	14\$:	JSR	PC,UPDSCR	:UPDATE THE SCRATCH INFORMATION	
227	024574	005237	002360		INC	LOCYL	:FINISHED WITH THIS CYLINDER,	
228	024600	005237	002304		INC	DESCYL	:GET THE NEXT CYLINDER IN THE SEQUENCE	
229	024604	023737	002360	002356	15\$:	CMP	LOCYL,HICYL	:STILL IN RANGE?
230	024612	101760			BLOS	13\$:IF <=, YES	
231	024614	000137	024142		JMP	2\$:GO BACK AND LOOK FOR ANOTHER SCRATCH	
232								
233	024620	062701	000016	16\$:	ADD	#<7*2>,R1	:NEXT BUFFER ENTRY, PLEASE	
234	024624	026637	177776	002356	17\$:	CMP	-2(SP),HICYL	:STILL IN RANGE?
235	024632	101402			BLOS	18\$:IF SO, TAKE THE BRANCH	
236	024634	000137	024200		JMP	4\$:GET THE NEXT TRACK AND LOOK FOR ANOTHER SCRATCH	
237								
238	024640	126137	000003	002245	18\$:	CMPB	3(R1),DESTRK+1	:TRACK MATCH ?
239	024646	001402			BEQ	19\$:IF SO, LOG THE DATA	
240	024650	103763			BLO	16\$:IF LOWER, KEEP MOVING THROUGH THE BUFFER	
241	024652	101346			BHI	14\$:IF HIGHER, GET THE TD FROM THE DISK, AND UPDATE IT	
242	024654	012705	000007	19\$:	MOV	#7,R5	:7 ITEMS / RECORD	
243	024660	012124		20\$:	MOV	(R1)+,(R4)+	:LOG THIS SCRATCH DATA	
244	024662	005305			DEC	R5	:ONE LESS ITEM / THIS RECORD	
245	024664	003375			BGT	20\$:KEEP GOING UNTIL R4 = 0	
246	024666	162701	000016		SUB	#<7*2>,R1	:BACK R1 UP TO THE TOP OF THIS RECORD	
247	024672	004737	024054		JSR	PC,COMPRS	:STRIP OUT THIS DEFECT, AND COMPRESS THE BUFFER	
248	024676	005237	002304		INC	DESCYL	:NEXT DESIRED CYLINDER PLEASE	
249	024702	005237	002360		INC	LOCYL	:ONE LESS CYLINDER TO-GO	
250	024706	020127	035656		CMP	R1,#ENDDF	:END OF DEFECT BUFFER YET ?	
251	024712	101734			BLOS	15\$:BR IF NO	
252	024714	000137	024142		JMP	2\$:GET THE NEXT CYLINDER ADDRESS	
253								
254	024720	012701	032250	21\$:	MOV	#DEFBUF,R1	:GET POINTER TO BEGINNING OF DEFECT BUFFER	
255	024724	012124		22\$:	MOV	(R1)+,(R4)+	:LOG THE NON SCRATCH RELATED DATA	
256	024726	005711			TST	(R1)	:VALID DATA ?	
257	024730	001375			BNE	22\$:KEEP GOING UNTIL R1 = 0	
258	024732	012637	002244		MOV	(SP)+,DESTRK	:POP STACK INTO DESTRK	
	024736	012637	002304		MOV	(SP)+,DESCYL	:POP STACK INTO DESCYL	
259	024742	000207			RTS	PC	: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
48
49
50
51
52
53

.....
: THIS MODULE IS USED BY THE SCANNER MODE OF OPERATION
: *
: INPUTS ARE: FROM R1
: *
: OUTPUTS ARE: DESCYL, DESTRK, TRACK DESCRIPTOR BUFFER CALLED 'TDWRD1'
: *
: THIS MODULE IS CALLED BY: MAIN
:

```
LOADTD:
MOV      RO,-(SP)          ;;PUSH RO ON STACK
MOV      #TDWRD1,R0       ;;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)+,(R0)+      ;;LOAD CYLINDER ADDRESS INTO TD WORD #1
MOV      (R1),DESTRK      ;;LOAD THE TRACK ADDRESS AND
CLRB     DESTRK           ;;THE SECTOR ADDRESS.
MOV      (R1)+,(R0)+      ;;LOAD TRACK/SECTOR ADRS INTO TD WORD #2
MOV      (R1)+,(R0)+      ;;LOAD TD WORD #3,
MOV      (R1)+,(R0)+      ;;TD WORD #4,
MOV      (R1)+,(R0)+      ;;TD WORD #5,
MOV      (R1)+,(R0)       ;;AND TD WORD #6.
TST      (R1)+            ;;GET OVER INTERNAL STATUS WORD IN TD LIST
MOV      (SP)+,R0         ;;POP STACK INTO R0
RTS      PC               ;;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,R0       ;;GET POINTER TO BEGINNING OF TD BUFFER
MOV      #TDTMP1,R1       ;;GET POINTER TO BEGINNING OF TD TEMP. 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)+,R0         ;;POP STACK INTO R0
RTS      PC
```

.....
: THIS MODULE IS USED TO RESTORE THE CONTENTS OF A BUFFER CALLED 'TDTMP1' WITH
: THE CONTENTS OF A BUFFER CALLED 'TDWRD1'.
:

54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97

```

025054 010046
025054 010146
025056 010146
025060 012700 030230
025064 012701 030200
025070 012021
025072 012021
025074 012021
025076 012021
025100 012021
025102 011011
025104 012601
025106 012600
025110 000207

025112 010046
025112 010046
025114 012700 030200
025120 013710 002304
025124 052720 030000
025130 013710 002244
025134 052720 000377
025140 012720 140000
025144 012720 140000
025150 012720 140000
025154 012710 140000
025160 012600
025162 000207
  
```

```

: *
: INPUTS ARE:   TDTMP1 - TDTMP6
:
: OUTPUTS ARE:  TDWRD1 - TDWRD6
: *
: THIS MODULE IS CALLED BY: MAIN
: *****
  
```

```

RESTTD:
MOV     R0,-(SP)      ;;PUSH R0 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)+,R0     ;;POP STACK INTO R0
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     R0,-(SP)      ;;PUSH R0 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)+,R0     ;;POP STACK INTO R0
RTS     PC           ;;TAKE THE RETURN NOW
  
```



```

1
2
3
4
5
6
7
8
9
10 025164 105737 002235 NEWDAT: TSTB FIRPAS ;SCANNER FIRST PASS ?
11 025170 001420 BEQ 4$ ;IF SO, DON'T RUN THIS MODULE
12 025172 010246 MOV R2,-(SP) ;SAVE R2
13 025174 010146 MOV R1,-(SP) ;AND SAVE R1
14 025176 013701 002232 MOV LASLOC,R1 ;GET POINTER TO BEGINNING OF LAST DEFECT
15 025202 012702 032250 MOV #DEFBUF,R2 ;GET POINTER TO BEGINNING OF DEFECT BUFFER
16 025206 012122 1$: MOV (R1)+,(R2)+ ;MOVE THE NEW DATA TO THE TOP OF THE BUFFER
17 025210 020116 CMP R1,(SP) ;DONE WITH NEW DATA ?
18 025212 103775 BLO 1$ ;IF LOWER, NO
19 025214 020227 035656 2$: CMP R2,#ENDDF ;END OF DEFECT BUFFER YET ?
20 025220 101002 BHI 3$ ;BR IF YES
21 025222 005022 CLR (R2)+ ;INITIALIZE THE UNUSED BUFFER LOCATIONS
22 025224 000773 BR 2$ ;DO UNTIL DONE
23 025226 005726 3$: TST (SP)+ ;POP STACK ONCE
24 025230 012602 MOV (SP)+,R2 ;RESTORE R2
25 025232 000207 4$: RTS PC ;NOW TAKE THE RETURN
26
27
28
29
30
31
32 025234 012701 032250 RESTOR: MOV #DEFBUF,R1 ;GET POINTER TO BEGINNING OF DEFECT BUFFER
33 025240 012703 035660 MOV #DBUFF,R3 ;GET POINTER TO BEGINNING OF DATA BUFFER
34 025244 032763 000400 000014 1$: BIT #BIT8,14(R3) ;WAS IT AN INTERPOLATED SCRATCH ?
35 025252 001406 BEQ 2$ ;BR IF NO
36 025254 062703 000016 ADD #<7*2>,R3 ;DO NOT RESTORE ENTRY BACK TO DEFECT BUFFER
37 025260 020327 117656 CMP R3,#ENDBUF ;DONE WITH BUFFER YET ?
38 025264 101767 BLOS 1$ ;BR IF NO
39 025266 000415 BR 5$
40
41 025270 005713 2$: TST (R3) ;ANY MORE ENTRIES ?
42 025272 001003 BNE 3$ ;BR IF YES
43 025274 010137 002232 MOV R1,LASLOC ;GET THE FIRST UNUSED BUFFER ADDRESS
44 025300 000410 BR 5$ ;AND GET OUT
45
46 025302 012704 000007 3$: MOV #7,R4 ;SEVEN ENTRIES / RECORD
47 025306 012321 4$: MOV (R3)+,(R1)+ ;RESTORE THE ORIGINAL BUFFER
48 025310 005304 DEC R4 ;ONE LESS ENTRY THIS RECORD
49 025312 003375 BGT 4$ ;DO UNTIL R4 = 0
50 025314 020127 035656 CMP R1,#ENDDF ;END OF DEFECT BUFFER YET ?
51 025320 101751 BLOS 1$ ;BR IF NO
52 025322 000207 5$: RTS PC
53
62
69
    
```

12
40
42
43
44
45
46
47
48
60
61
62
74
75
76

025324
025324 000167
025326 000000
025330
025330 104425

.SBTTL REPORT CODING SECTION
:++
: THE REPORT CODING SECTION CONTAINS THE
: "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
:--
L\$RPT::
.WORD JSJMP
.WORD L10007-2-
.EVEN
L10007:
TRAP CSRPT

1
2
3
4
5
6
7
8 025332
9 025332 000000
10 025334 177777
11 025336 000006
13

.SBTTL PROTECTION TABLE

:++
: THIS TABLE IS USED BY THE RUNTIME SERVICES
: TO PROTECT THE LOAD MEDIA.
:--

L\$PROT::

0

-1

6

:P-TABLE OFFSET OF CSR
:NOT A MASSBUS DEVICE
:P-TABLE OFFSET DRIVE #

```

1
2
3
4
5
6
7
8 025340
9
10 025340 104433
11
12 025342 012700 000036
   025346 104447
13
14 025350 103002
15
16 025352 104432
   025354 000756
17 025356
18 025356 012700 000037
   025362 104447
19
20 025364 103414
21
22 025366 012700 000034
   025372 104447
23
24 025374 103002
25 025376 000137 026044
26 025402
27 025402 012700 000040
   025406 104447
28
29 025410 103402
30 025412 000137 026010
31 025416 005037 002114
32 025422 005227 177777
33 025426 001010
34 025430
35 025430 012746 004647
   025434 012746 000001
   025440 010600
   025442 104417
   025444 062706 000004
36 025450
37 025450 104443
   025452 000406
   025454 002160
   025456 000052
   025460 006067
   025462 000007
   025464 000000
   025466 000006
   025470
39 025470 005737 002160
40 025474 001410
41 025476 023727 002160 000002
    
```

```

.SBTTL INITIALIZE SECTION

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

LSINIT::

                                ;RESET THE WORLD
                                ;CONTINUE COMMAND ?
TRAP CSRESET
MOV #EF.CON,RO
TRAP CSREFG
                                ;GO TO 1$ IF NO
BCC 1$
                                ;EXIT IF YES
TRAP C$EXIT
.WORD L10011-
1$:
MOV #EF.RES,RO
TRAP CSREFG
                                ;GO TO 3$ IF YES
BCS 3$
                                ;POWER FAIL START ?
MOV #EF.PWR,RO
TRAP CSREFG
                                ;GO TO 2$ IF NO
BCC 2$
JMP 15$
                                ;GO TO 15$ IF YES
                                ;START COMMAND ?
2$:
MOV #EF.STA,RO
TRAP CSREFG
                                ;GO TO 3$ IF YES
BCS 3$
JMP 13$
                                ;GO TO 13$ IF NO
3$:
CLR L$TEST
                                ;CLEAR TEST NUMBER
4$:
INC #-1
                                ;FIRST TIME THRU HERE ?
BNE 6$
                                ;BR IF NO
5$:
                                ;PRINT THE HELP MESSAGE
MOV #FRMT50,-(SP)
MOV #1,-(SP)
MOV SP,RO
TRAP CSPNTF
ADD #4,SP
6$:
                                ;PRINT 'OPTIONS (6=HELP) (?) 0 ?'
TRAP C$GMAN
BR 10000$
.WORD OPTION
.WORD T$CODE
.WORD MSG7
.WORD 7
.WORD T$LOLIM
.WORD T$HILIM
10000$:
TST OPTION
BEQ 7$
CMP OPTION,#2
                                ;FORMAT OPTION ?
                                ;BR IF YES
                                ;SCAN OPTION ?
    
```



```
42 025504 001404          BEQ      7$
43 025506 023727 002160 000004  CMP      OPTION,#4
44 025514 001012          BNE      8$
45 025516          7$:
46
47
48 025516 104443          TRAP    CS$GMAN
   025520 000404          BR      10001$
   025522 002152          .WORD  CONTIN
   025524 000120          .WORD  T$CODE
   025526 006155          .WORD  MSG25
   025530 000001          .WORD  1
   025532          10001$:
49 025532 005737 002152          TST    CONTIN
50 025536 001001          BNE    8$
51 025540 104444          TRAP   CS$DCLN
53 025542 023727 002160 000006 8$:  CMP    OPTION,#6
54 025550 001727          BEQ    5$
55 025552 023727 002160 000004  CMP    OPTION,#4
56 025560 002110          BGE    12$
57
58 025562 005037 002156          CLR    DRVPAR
59
60 025566 104443          TRAP   CS$GMAN
   025570 000404          BR      10002$
   025572 002156          .WORD  DRVPAR
   025574 000130          .WORD  T$CODE
   025576 006037          .WORD  MSG18
   025600 000001          .WORD  1
   025602          10002$:
61 025602 005737 002156          TST    DRVPAR
62 025606 001440          BEQ    9$
63
64 025610 104443          TRAP   CS$GMAN
   025612 000406          BR      10003$
   025614 002204          .WORD  MINCYL
   025616 000052          .WORD  T$CODE
   025620 006135          .WORD  MSG15
   025622 001777          .WORD  1777
   025624 000000          .WORD  T$LOLIM
   025626 001166          .WORD  T$HILIM
   025630          10003$:
65
66 025630 104443          TRAP   CS$GMAN
   025632 000406          BR      10004$
   025634 002206          .WORD  MAXCYL
   025636 000052          .WORD  T$CODE
   025640 006145          .WORD  MSG16
   025642 001777          .WORD  1777
   025644 000000          .WORD  T$LOLIM
   025646 001166          .WORD  T$HILIM
   025650          10004$:
67
68 025650 104443          TRAP   CS$GMAN
   025652 000406          BR      10005$
   025654 002210          .WORD  MINTRK
   025656 000052          .WORD  T$CODE

:BR IF YES
:MODIFY OPTION ?
:BR IF NO
:PRINT '! CUSTOMER DATA WILL BE OVERWRITTEN!'
:-----
:CONTINUE ?

:CONTINUE ?
:BR IF YES

:PRINT HELP MESSAGE ?
:BR IF YES
:WHAT IS OPTION ?
:BR IF MODIFY OR WRITE FE-2
:ELSE, OPTION IS FORMAT, VERIFY, SCAN OR LIST
:DEFAULT TO 'N' FOR CHANGE PARAMETERS
:PRINT 'CHANGE DRIVE PARAMETERS (L) N ?'

:CHANGE DRIVE PARAMETERS ?
:BR IF NO
:PRINT 'MIN CYL (D) 0 ?'

:PRINT 'MAX CYL (D) 630 ?'

:PRINT 'MIN TRK (D) 0 ?'
```

```

025660 006115      .WORD  MSG13
025662 000037      .WORD  37
025664 000000      .WORD  T$LOLIM
025666 000037      .WORD  T$HILIM
025670
10005$:
69
70 025670 104443      TRAP  C$GMAN      ;PRINT 'MAX TRK (D) 31 ?'
025672 000406      BR    10006$
025674 002212      .WORD  MAXTRK
025676 000052      .WORD  T$CODE
025700 006125      .WORD  MSG14
025702 000037      .WORD  37
025704 000000      .WORD  T$LOLIM
025706 000037      .WORD  T$HILIM
025710
10006$:
71
72 025710 005737 002160 9$:  TST  OPTION      ;FORMAT OPTION ?
73 025714 001006      BNE  10$         ;BR IF NO
74
75 025716 104443      TRAP  C$GMAN
025720 000404      BR    10007$
025722 002164      .WORD  NOWRCK
025724 000130      .WORD  T$CODE
025726 006310      .WORD  MSG17
025730 000001      .WORD  1
025732
10007$:
76
77 025732 023727 002160 000002 10$:  CMP  OPTION,#2   ;SCAN OPTION ?
78 025740 001006      BNE  11$         ;BR IF NO
83
84
85 025742 104443      TRAP  C$GMAN
025744 000404      BR    10010$
025746 002172      .WORD  ENWTTD
025750 000130      .WORD  T$CODE
025752 006403      .WORD  MSG19
025754 000001      .WORD  1
025756
10010$:
86
87 025756 023727 002160 000003 11$:  CMP  OPTION,#3   ;LIST OPTION ?
88 025764 001006      BNE  12$         ;BR IF NO
89
90 025766 104443      TRAP  C$GMAN
025770 000404      BR    10011$
025772 002202      .WORD  LISHDR
025774 000130      .WORD  T$CODE
025776 006462      .WORD  MSG23
026000 000001      .WORD  1
026002
10011$:
91 026002 012737 177777 002404 12$:  MOV  #-1,UNIT    ;INITIALIZE FOR UNIT 0 ON START
92 026010 005237 002404 13$:  INC  UNIT        ;INCREMENT TO NEXT UNIT
93 026014 004737 014152      JSR  PC,TABLD    ;LOAD THE HARDWARE P-TABLES
94 026020 023737 002404 002012  CMP  UNIT,L$UNIT ;OUT OF UNITS TO TEST ?
95 026026 002403      BLT  14$         ;BR IF NO
96 026030 104444      TRAP  C$DCLN
97 026032 104432      TRAP  C$EXIT
026034 000276      .WORD  L10011-
  
```



```

98 026036
99 026036 012700 000340
   026042 104441
100
101
102
103 026044 005037 002242
104 026050 005037 002162
105 026054 005037 002166
106 026060 005037 002170
107 026064 005037 002174
108 026070 005037 002200
109 026074 005037 002154
110 026100 010146
111 026102 012701 002234
112 026106 105021
113 026110 020127 002237
114 026114 101774
115 026116 012701 002244
116 026122 005021
117 026124 020127 002304
118 026130 101774
119 026132 012701 002564
120
121
122
123 026136 005737 002160
124 026142 001003
125 026144 005237 002162
126 026150 000437
127 026152 023727 002160 000001 18$:
128 026160 001003
129 026162 005237 002166
130 026166 000433
131 026170 023727 002160 000002 19$:
132 026176 001003
133 026200 005237 002170
134 026204 000442
135 026206 023727 002160 000003 20$:
136 026214 001003
137 026216 005237 002200
138 026222 000431
139 026224 023727 002160 000004 21$:
140 026232 001003
141 026234 005237 002174
142 026240 000413
143 026242 005237 002154
144 026246 000403
145
146
147
148 026250 012721 000075
149 026254 000411
150
151
152
153 026256 012721 000075
  
```

```

14$:
MOV #PRI07,R0 ;SET PRIORITY TO 7
TRAP C$SPRI

;RESET SOME COMMON TABLES AND PARAMETERS

15$:
CLR SUPRSS ;ALLOW ERROR MESSAGE TO BE OUTPUT
CLR FORMT ;DISABLE FORMAT OPTION
CLR VRIFY ;DISABLE VERIFY OPTION
CLR SCANR ;DISABLE SCAN OPTION
CLR MODTD ;DISABLE MODIFY OPTION
CLR LIST ;DISABLE LIST OPTION
CLR WRTFE2 ;DISABLE WRITE FE-2 OPTION
MOV R1,-(SP) ;PUSH R1 ON STACK
MOV #CEMODE,R1 ;LOAD THE TOP OF THE CONTROL FILE
16$:
CLRB (R1)+ ;AND RESET IT
CMP R1,#INTLEV ;FINISHED?
BLOS 16$ ;IF NOT >, NO
MOV #DESTRK,R1 ;GET THE TOP OF THE STATUS FILE
17$:
CLR (R1)+ ;AND RESET IT
CMP R1,#DESCYL ;FINISHED?
BLOS 17$ ;IF NOT >, NO!
MOV #CMDQUE,R1 ;SET R1 = THE TOP OF THE COMMAND QUEUE

;FIND WHAT OPTION WAS SPECIFIED

TST OPTION ;FORMAT OPTION ?
BNE 18$ ;BR IF NO
INC FORMT ;SET FORMAT OPTION
BR 23$
18$:
CMP OPTION,#1 ;VERIFY OPTION ?
BNE 19$ ;BR IF NO
INC VRIFY ;SET VERIFY OPTION
BR 24$
19$:
CMP OPTION,#2 ;SCAN OPTION ?
BNE 20$ ;BR IF NO
INC SCANR ;SET SCAN OPTION
BR 28$
20$:
CMP OPTION,#3 ;LIST OPTION ?
BNE 21$ ;BR IF NO
INC LIST ;SET LIST OPTION
BR 27$
21$:
CMP OPTION,#4 ;MODIFY OPTION ?
BNE 22$ ;BR IF NO
INC MODTD ;SET MODIFY OPTION
BR 25$
22$:
INC WRTFE2 ;SET WRITE FE-2 OPTION
BR 24$

;ENTER HERE TO FORMAT HEADERS & DATA

23$:
MOV #RD TD,(R1)+ ;LOAD THE READ TD OPERATION
BR 26$

;ENTER HERE TO VERIFY TD(S) & HEADER(S)

24$:
MOV #RD TD,(R1)+ ;LOAD THE READ TD OPERATION
  
```

```

154 026262 012721 000073      MOV    #RDHD,(R1)+    ;LOAD THE READ HEADER AND DATA FUNCTION
155 026266 000411      BR     28$
156
157                          ;ENTER HERE TO MODIFY TD(S)
158
159 026270 005037 002402      25$:  CLR    MORETD      ;DEFAULT TO NO MORE TD'S TO MODIFY
160                          ;THE READ TD COMMAND IS IMBEDDED IN THE
161                          ;'INPUTD' SUBROUTINE
162 026274 012721 000065      MOV    #WRTTD,(R1)+   ;LOAD THE WRITE TD OPERATION
163 026300 012721 000063      26$:  MOV    #FMTRK,(R1)+ ;LOAD THE FORMAT TRACK OPERATION
164 026304 000402      BR     28$
165
166                          ;ENTER HERE TO LIST TD(S) WITH DEFECTS
167
168 026306 012721 000075      27$:  MOV    #RDTD,(R1)+ ;LOAD THE READ TD OPERATION
169
170                          ;ENTER HERE TO SCAN FOR NEW DEFECTS
171
172 026312 022701 002576      28$:  CMP    #CMDQUE+10.,R1 ;QUEUE FULL?
173 026316 103402      BLO    29$            ;TAKE BRANCH IF SO
174 026320 005021      CLR    (R1)+         ;ZERO THE BALANCE OF THE QUEUE
175 026322 000773      BR     28$           ;AND DO UNTIL QUEUE IS DONE
176 026324 012601      29$:  MOV    (SP)+,R1    ;;POP STACK INTO R1
177
201
202 026326 104432      TRAP   C$EXIT
203 026330 000002      .WORD L10011-.
204
205
206
207
208
209
210
211
212
213
214
215
216
217 026332 104411      L10011: TRAP   C$INIT
  
```


1
2
3
4
5
6
7
8
9
10 026334
11
18 026334
026334 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
2
3
4
5
6
7
8 026336
9
10 026336 012700 000340
11 026342 104441
12 026344 012777 000040 154062
13 026352 013777 002420 154054
14 026360 012746 002642
15 026364 012746 000001
16 026370 010600
17 026372 104417
18 026374 062706 000004
19 026400 005737 002114
20 026404 001002
21 026406 104432
22 026410 000232
23
24 026412 012701 000001
25 026416 013746 002274
26 026422 010146
27 026424 012746 003516
28 026430 012746 000003
29 026434 010600
30 026436 104417
31 026440 062706 000010
32 026444 005201
33
34 026446 013746 002276
35 026452 010146
36 026454 012746 003516
37 026460 012746 000003
38 026464 010600
39 026466 104417
40 026470 062706 000010
41 026474 005201
42
43 026476 013746 002300
44 026502 010146
45 026504 012746 003516
46 026510 012746 000003
47 026514 010600
48 026516 104417
49 026520 062706 000010
50 026524 005737 002302
51 026530 001414
52
53 026532 013746 002302
54 026536 012746 006016

```

```

.SBTTL CLEANUP CODING SECTION

:
: **
: THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
: AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
:
: --

L$CLEAN::

;SET PRIORITY TO 7

MOV #PRI07,R0
TRAP C$SPRI

MOV #CLR,@RPCS2 ;MASSBUS INIT TO CLEAR IMPENDING INTERRUPTS
MOV DRVNO,@RPCS2 ;GET DRIVE NUMBER
;CR-LF

MOV #CRLF,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #4,SP

TST L$TEST ;DID PROGRAM ENTER TEST 1 ?
BNE 1$ ;BR IF YES
TRAP C$EXIT
.WORD L10013-

1$: MOV #1,R1 ;R1 = 1
;PRINT 'TRACKS WITH 1 DEFECTS='

MOV DEF1,-(SP)
MOV R1,-(SP)
MOV #FRMT12,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
INC R1 ;R1 = 2
;PRINT 'TRACKS WITH 2 DEFECTS='

MOV DEF2,-(SP)
MOV R1,-(SP)
MOV #FRMT12,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
INC R1 ;R1 = 3
;PRINT 'TRACKS WITH 3 DEFECTS='

MOV DEF3,-(SP)
MOV R1,-(SP)
MOV #FRMT12,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C$PNTF
ADD #10,SP
TST DEF4 ;WERE THERE ANY TRACKS WITH 4 DEFECTS ?
BEQ 2$ ;BR IF NO
;PRINT '** WARNING ** THERE WERE X. TRACK(S)
; WITH 4 DEFECTS'

MOV DEF4,-(SP)
MOV #WARN,-(SP)

```


	026542	012746	005156		MOV	#FRMT62,-(SP)	
	026546	012746	000003		MOV	#3,-(SP)	
	026552	010600			MOV	SP,R0	
	026554	104417			TRAP	C\$PNTF	
	026556	062706	000010		ADD	#10,SP	
33	026562			2\$:			;PRINT 'TOTAL DEFECTS FOUND= X.'
34	026562	013746	002246		MOV	DEFCNT,-(SP)	
	026566	012746	003564		MOV	#FRMT13,-(SP)	
	026572	012746	000002		MOV	#2,-(SP)	
	026576	010600			MOV	SP,R0	
	026600	104417			TRAP	C\$PNTF	
	026602	062706	000006		ADD	#6,SP	
35							;PRINT 'DONE, RETRIES MADE= X., ERRORS DETECTED= X'
36	026606	013746	002260		MOV	ERTTL,-(SP)	
	026612	013746	002262		MOV	ERRTTL,-(SP)	
	026616	012746	003423		MOV	#FRMT11,-(SP)	
	026622	012746	000003		MOV	#3,-(SP)	
	026626	010600			MOV	SP,R0	
	026630	104417			TRAP	C\$PNTF	
	026632	062706	000010		ADD	#10,SP	
37							
46							
47	026636	104432			TRAP	C\$EXIT	
	026640	000002			.WORD	L10013-	
48							
60					.EVEN		
61							
62	026642			L10013:			
	026642	104412			TRAP	C\$CLEAN	

1
2
3
4
5
6
7
8
9
18
19
20
32
33
34

.SBTTL DROP UNIT SECTION

::+
: THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
: TO NO LONGER BE TESTED.
:--

026644

L\$DU::

026644 000167
026646 000000

.WORD J\$JMP
.WORD L10014-2-

.EVEN

026650
026650 104453

L10014:
TRAP C\$DU

1
2
3
4
5
6
7
8
9 026652
10
19
20 026652 000167
026654 000000
21
33
34
35 026656
026656 104452

.SBTTL ADD UNIT SECTION

::++
: THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
: TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
: TO THE TEST CYCLE.
:--

L\$AU::

.WORD JSJMP
.WORD L10015-2-

.EVEN

L10015:
TRAP C\$AU

```

2          .SBTTL  HARDWARE TESTS
13         .SBTTL  TEST 1: MAIN PROGRAM
49
51
52         :++
53         : TEST #1      MAIN FORMAT, VERIFY AND LIST OPERATIONS START HERE
54         :--
61
67
68 026660 T1::
69 026660 005037 002264          CLR      ERRMSK          :RESET NO ERROR STATUS AND
70 026664 005037 002266          CLR      RTYCNT          :RESET RETRY COUNT TO 0
71 026670 012737 000005 002222  MOV      #5,ERRMAX       :LOAD THE ERROR LIMIT
72 026676 004737 013460          JSR      PC,DRVINI       :INITIALIZE THE DEVICE
73 026702 013737 002204 002304  MOV      MINCYL,DESCYL   :LOAD THE MINIMUM DESIRED CYLINDER
74 026710 013737 002210 002244  MOV      MINTRK,DESTRK  :LOAD THE MINIMUM DESIRED TRACK
75 026716 000337 002244          SWAB     DESTRK          :TRACK ADDRESS IS THE HIGH BYTE
76 026722 005737 002256          TST     SOFSW           :DID DRIVE INITIALIZE OK?
77 026726 001402          BEQ     1$              :BR IF YES
78 026730 104432          TRAP   C$EXIT
        026732 000760          .WORD  L10016-.
79
80 026734 005737 002154 1$:      TST     WRTFE2          :ARE WE WRITTING 2ND FE CYLINDER ?
81 026740 001404          BEQ     2$              :BR IF NO
82 026742 004737 020120          JSR     PC,FETWO        :GO FORMAT FE CYLINDER 2
83 026746 104432          TRAP   C$EXIT
        026750 000742          .WORD  L10016-.
84
85 026752 004737 017164 2$:      JSR     PC,RDBSF        :READ BAD SECTOR FILE (DEC 144 FILE)
86 026756 005737 002170          TST     SCANR          :ARE WE DOING DISK SCAN ?
87 026762 001075          BNE    12$             :BR IF YES
88
89          :FORMAT, VERIFY & MODIFY OPTIONS START HERE
90
91 026764 3$:
92 026764 104422          TRAP   C$BRK           :>>>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<<<<<<<
93 026766 012705 002564          MOV     #CMDQUE,R5     :GET THE LIST OF COMMANDS
94 026772 022715 000075          CMP     #RDTD,(R5)     :IS THE COMMAND A READ TD ?
95 026776 001004          BNE    4$              :BR IF NO
96 027000 004737 020466          JSR     PC,READTD      :CALL THE READ TD SUBROUTINE
97 027004 000450          BR     10$             :RETURN HERE IF EXCEEDED RETRY LIMIT
98 027006 000407          BR     5$              :ELSE RETURN HERE, NO ERROR
99
100 027010 021527 000065 4$:      CMP     (R5),#WRTTD     :IS THE COMMAND A WRITE TD ?
101 027014 001005          BNE    6$              :BR IF NO
102 027016 104450          TRAP   C$MANI
103 027020 103003          BCC    6$              :
104 027022 004737 011370          JSR     PC,INPUTD      :CALL THE INPUT TD SUBROUTINE
105
106 027026 005725 5$:      TST     (R5)+          :POINT TO NEXT COMMAND IN QUEUE
107 027030 005715 6$:      TST     (R5)          :ANY MORE COMMANDS IN QUEUE ?
108 027032 001417          BEQ    8$              :BR IF NO
109 027034 012537 002370          MOV     (R5)+,FUNCTN   :THIS IS A FORMAT OR VERIFY COMMAND
110 027040 004737 013302 7$:      JSR     PC,MAKTRK      :NOW EXECUTE THE COMMAND
111 027044 000430          BR     10$            :RETURN HERE IF EXCEEDED RETRY LIMIT
112
113 027046 022737 000073 002370  CMP     #RDHD,FUNCTN    :IS THIS A VERIFY COMMAND ?
    
```


114	027054	001006			BNE	8\$:BR IF NO
115	027056	004737	017064		JSR	PC,VERIFY			:ANY MIS-COMPARES DURING HEADER VERIFY ?
116	027062	001403			BEQ	8\$:BR IF NO
117	027064	004737	012752		JSR	PC,ERRORS			:REPORT THE ERROR
118	027070	000416			BR	10\$			
119									
120	027072	005737	002200	8\$:	TST	LIST			:IS LIST OPTION ENABLED ?
121	027076	001402			BEQ	9\$:BR IF NO
122	027100	004737	020742		JSR	PC,LISTDS			:CALL SUBROUTINE TO LIST THE TD(S) WITH DEFECTS
123	027104	004737	017750	9\$:	JSR	PC,CNTDEF			:INCLUDE THIS DATA IN THE DEFECT COUNT
124	027110	005737	002174		TST	MODTD			:ARE WE IN MODIFY TD MODE ?
125	027114	001404			BEQ	10\$:BR IF NO
126	027116	005737	002402		TST	MORETD			:DO WE HAVE ANYMORE TD'S TO MODIFY ?
127	027122	001320			BNE	3\$:BR IF YES
128	027124	000406			BR	11\$			
129									
130	027126	004737	017720	10\$:	JSR	PC,SPIRAL			:GET THE NEXT SEQUENTIAL DISK ADDRESS
131	027132	023737	002304	002206	CMP	DESCYL,MAXCYL			:DID WE OVERFLOW?
132	027140	101711			BLOS	3\$:BR IF NO
133	027142	004737	015316	11\$:	JSR	PC,SESORT			:SORT THE BAD SECTOR FILE
134	027146	004737	017436		JSR	PC,WRTBSF			:WRITE BAD SECTOR FILE (DEC 144 FILE)
135	027152	104432			TRAP	C\$EXIT			
	027154	000536			.WORD	L10016-			
136	027156			12\$:					

```

1
2
3
4
5 027156 105037 002236
6 027162 012702 031260
7
8 027166 005712
9 027170 100421
10 027172 022222
11 027174 020227 032246
12 027200 101772
13
14
15 027202 012746 006016
    027206 012746 005237
    027212 012746 000002
    027216 010600
    027220 104417
    027222 062706 000006
16 027226 112737 177777 002241
17
18 027234 012737 177777 002242
19 027242 105037 002235
20 027246 012737 032250 002232
21 027254 012701 032250
22 027260 005021
23 027262 020127 117656
24 027266 101774
25 027270 012701 032250
26
27 027274
28 027274 104422
29 027276 004737 021572
30 027302 000407
31
32
33 027304 004737 021726
34 027310 000404
35
36
37 027312 004737 022220
38 027316 000401
39 027320 000405
40
41 027322 004737 012752
42 027326 012737 177777 002242
43
44 027334 105737 002235
45 027340 001403
46 027342 004737 014316
47 027346 000402
48 027350 004737 017720
49
50 027354 023727 002250 000200
51 027362 101017
52 027364 105737 002235

```

```

:++
: TEST #1      MAIN SCAN OPERATION STARTS HERE
:--
SCAN:  CLRB   SCANIT          ;RESET SCAN ITERATION COUNT
      MOV     #USRBUF+8.,R2    ;GET POINTER TO BEGINNING OF USER BUFFER
1$:    TST     (R2)             ;TO GET OVER 2 S/N WRDS AND 2 ALL 0'S WRDS.
      BMI     2$               ;IS THERE ANY ROOM IN THE USR BSF ?
      CMP     (R2)+,(R2)+     ;BR IF YES
      CMP     R2,#ENDUSR     ;NEXT POINTER ENTRY, PLEASE
      BLOS   1$              ;END OF USR BUFFER YET ?
      PRINT  '** WARNING ** USR BAD SECTOR FILE
      BUFFER IS FULL NOW'
      MOV     #WARN,-(SP)
      MOV     #FRMT63,-(SP)
      MOV     #2,-(SP)
      MOV     SP,R0
      TRAP   C$PNTF
      ADD    #6,SP
16:    MOV     #-1,NOROOM      ;MARK THE BAD SECTOR FILE AS FULL
18:    MOV     #-1,SUPRSS     ;SUPPRESS THE ERROR MESSAGE OUTPUT
      CLR    FIRPAS          ;SET FIRST PASS OF SCAN
      MOV     #DEFBUF,LASLOC ;SAVE LOCATION OF LAST DEFECT IN BUFFER
      MOV     #DEFBUF,R1     ;GET POINTER TO BEGINNING OF DEFECT BUFFER
22:    CLR     (R1)+         ;INITIALIZE DEFECT AND DATA BUFFERS
      CMP     R1,#ENDBUF     ;DONE BUFFERS YET ?
      BLOS   3$              ;BR IF NO
      MOV     #DEFBUF,R1     ;GET POINTER TO BEGINNING OF DEFECT BUFFER
27:    TRAP   C$BRK          ;>>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<<<<<<<
29:    JSR    PC,TDSCAN      ;LOOK FOR A CORRUPTED TD
      BR     5$              ;RETURN HERE IF ERROR
                               ;ELSE RETURN HERE, NO ERROR
33:    JSR    PC,HDSCAN      ;LOOK FOR A CORRUPTED HEADER
      BR     5$              ;RETURN HERE IF ERROR
                               ;ELSE RETURN HERE, NO ERROR
37:    JSR    PC,DASCAN      ;LOOK FOR DATA DEFECTS
      BR     5$              ;RETURN HERE IF ERROR
      BR     6$              ;ELSE RETURN HERE, NO ERROR
42:    JSR    PC,ERRORS     ;REPORT ERROR, IF SUPRSS=0
      MOV     #-1,SUPRSS     ;SUPPRESS THE ERROR MESSAGE OUTPUT
44:    TSTB   FIRPAS         ;ARE WE IN FIRST PASS OF SCAN ?
      BEQ    7$              ;BR IF YES
      JSR    PC,GETNEX     ;GET THE NEXT DISK ADDRESS IN DEFECT BUFFER
      BR     8$              ;AND GO-ON
48:    JSR    PC,SPIRAL     ;GET THE NEXT SEQUENTIAL DISK ADDRESS
50:    CMP     NEWCNT,#128.  ;IS DEFECT BUFFER FULL ?
      BMI     10$           ;BR IF YES
      TSTB   FIRPAS         ;ARE WE IN FIRST PASS OF SCAN ?

```



```

53 027370 100405          BMI      9$          :BR IF NO
54 027372 023737 002304 002206  CMP      DESCYL,MAXCYL :DONE ALL CYLINDERS YET ?
55 027400 101735          BLOS    4$          :BR IF NO
56 027402 000407          BR      10$
57
58 027404 023737 002252 002232 9$:  CMP      TBLPTR,LASLOC  :DONE CHECKING THE CORRECTED DEFECTS ?
59 027412 103730          BLO     4$          :BR IF NO
60 027414 023701 002232  CMP      LASLOC,R1     :DID WE FIND ANY NEW DEFECTS ?
61 027420 001522          BEQ     20$          :BR IF NO
62
63 027422 004737 025164          10$:  JSR     PC,NEWDAT      :GO OPTIMIZE BUFFER
64 027426 013746 002304          MOV     DESCYL,-(SP)   :PUSH DESCYL ON STACK
   027432 013746 002244          MOV     DESTRK,-(SP)  :PUSH DESTRK ON STACK
65 027436 012701 035660          MOV     #DBUFF,R1    :GET TOP OF DATA BUFFER
66 027442 005021          11$:  CLR     (R1)+         :AND INITIALIZE IT
67 027444 020127 117656          CMP     R1,#ENDBUF   :DONE BUFFER YET ?
68 027450 101774          BLOS   11$          :BR IF NO
69 027452 004737 024104          JSR     PC,FINSCR    :LOOK FOR SCRATCHES IN THIS AREA;
70                                     :PUT THE CONTENTS OF 'DEFBUF' (INCLUDING ANY
71                                     :INTERPOLATED SCRATCHES) INTO 'DBUFF'.
72 027456 005737 035660          TST     DBUFF        :DID WE HAVE ANY DEFECTS ?
73 027462 001001          BNE    12$          :IF NOT ZERO, YES
74 027464 000506          BR     21$
75
76 027466 012737 035660 002230 12$:  MOV     #DBUFF,BADDR  :GET POINTER TO BEGINNING OF DATA BUFFER
81 027474 004737 021252          JSR     PC,NEWLST    :PRINT NEW DEFECTS, IF ANY
82
83                                     :SEE IF USER WANTS TO RE-WRITE TD(S) OF NEW DEFECTS
84
85 027500 005737 002250          13$:  TST     NEWCNT      :ANY NEW DEFECTS FOUND ?
86 027504 001476          BEQ    21$          :IF ZERO, NO
87 027506 005737 002172          TST     ENWTTD      :IS WRITE TD(S) ENABLED ?
88 027512 001473          BEQ    21$          :BR IF NO
89 027514 013700 002222          MOV     ERRMAX,R0   :SETUP RETRY COUNT
90 027520 013701 002230          MOV     BADDR,R1    :GET ADDRESS OF BUFFER
91 027524 004737 024744          14$:  JSR     PC,LOADTD   :LOAD THE TD FOR THE CORRECT TRACK
92 027530 004737 025016          JSR     PC,SAVETD   :SAVE TD WRITE BUFFER JUST IN CASE THERE
93                                     :IS RETRY ATTEMPT.
94 027534 004737 025054          15$:  JSR     PC,RESTTD   :NOW RESTORE THE TD BUFFER
95 027540 004737 020222          JSR     PC,WRITTD   :CALL THE WRITE TD SUBROUTINE
96 027544 000417          BR     18$         :RETURN HERE IF EXCEEDED RETRY LIMIT
97                                     :ELSE RETURN HERE, NO ERROR
98 027546 004737 020466          JSR     PC,READTD   :CALL THE READ TD SUBROUTINE
99 027552 000401          BR     16$         :RETURN HERE IF EXCEEDED RETRY LIMIT
100 027554 000403          BR     17$         :ELSE RETURN HERE, NO ERROR
101 027556 005300          16$:  DEC     R0          :DID WE EXCEED RETRY ?
102 027560 001411          BEQ    18$         :BR IF YES
103 027562 000764          BR     15$         :TRY WRITTING TD AGAIN
104
105 027564 012737 000063 002370 17$:  MOV     #FMTRK,FUNCTN :LOAD THE FORMAT COMMAND FOR THE DRIVER
106 027572 004737 013302          JSR     PC,MAKTRK   :NOW EXECUTE THE COMMAND
107 027576 000402          BR     18$         :RETURN HERE IF EXCEEDED RETRY LIMIT
108                                     :ELSE RETURN HERE, NO ERROR
109 027600 004737 017750          18$:  JSR     PC,CNTDEF   :COUNT THESE NEW DEFECTS IN THE OVERALL COUNTER
110 027604 005711          TST     (R1)        :GOOD DATA FOR THIS ENTRY?
111 027606 001403          BEQ    19$         :IF ZERO, WE'RE DONE
112 027610 020127 117656          CMP     R1,#ENDBUF  :AT END OF THE SCRATCH BUFFER?
  
```

```

113 027614 101743
114 027616 19$: BLOS 14$ ;BR IF NO
    027616 012637 002244 MOV (SP)+,DESTRK ;;POP STACK INTO DESTRK
    027622 012637 002304 MOV (SP)+,DESCYL ;;POP STACK INTO DESCYL
115
116 027626 005037 002250 CLR NEWCNT ;RESET THE LOCAL DEFECT COUNTER
117 027632 004737 025234 JSR PC,RESTOR ;NOW RESTORE THE ORIGINAL DEFECT BUFFER;
118 ;PUT THE CONTENTS OF 'DBUFF' (EXCLUDING THE
119 ;INTERPOLATED SCRATCHES) INTO 'DEFBUF'.
120 027636 012737 032250 002252 MOV #DEFBUF,TBLPTR ;GET POINTER TO BEGINNING OF DEFECT BUFFER
121 027644 004737 014316 JSR PC,GETNEX ;AND GET FIRST DATA-SET
122 027650 013701 002232 MOV LASLOC,R1 ;RESTORE POINTER THE INPUT BUFFER
123 027654 112737 177777 002235 MOVB #-1,FIRPAS ;INDICATE 2ND PASS OF SCANNER
124 027662 000137 027274 JMP 4$ ;DO SOME MORE SCANNING
125
126 027666 023737 002304 002206 20$: CMP DESCYL,MAXCYL ;DID WE DO ALL CYLINDERS?
127 027674 101002 BHI 21$ ;IF HIGHER, WE'RE DONE
128 027676 000137 027234 JMP 2$ ;DO SOME MORE CYLINDERS
129
130 027702 004737 015316 21$: JSR PC,SESORT ;SORT THE BAD SECTOR FILE
131 027706 004737 017436 JSR PC,WRTBSF ;WRITE BAD SECTOR FILE (DEC 144 FILE)
132
138
150 .EVEN
151
152 027712 104401 L10016: TRAP C$ETST
153 027712
  
```


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.
:--
  
```

```

53 027714 000022
    027716
54
55 027716 000031
    027720 027762
    027722 160000
    027724 177777
56
57 027726 001031
    027730 027775
    027732 000000
    027734 000377
58
59 027736 002032
    027740 030011
    027742 000340
    027744 000000
    027746 000007
60
61 027750 003032
    027752 030022
    027754 000007
    027756 000000
    027760 000007
62 027762
63
67 027762 122 120 103
68 027775 126 105 103
69 030011 102 122 040
70 030022 104 122 111
  
```

L\$HARD:: .WORD L10017-L\$HARD/2

```

        .WORD T$CODE
        .WORD MSG1
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD MSG4
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD MSG5
        .WORD 340
        .WORD T$LLOLIM
        .WORD T$HILIM
        .WORD T$CODE
        .WORD MSG6
        .WORD /
        .WORD T$LLOLIM
        .WORD T$HILIM
        .EVEN
  
```

;PRINT 'RPCS1 ADRS?'

;PRINT 'VECTOR ADRS?'

;PRINT 'BR LEVEL?'

;PRINT 'DRIVE #?'

L10017:

```

MSG1: .ASCIZ /RPCS1 ADRS/
MSG4: .ASCIZ /VECTOR ADRS/
MSG5: .ASCIZ /BR LEVEL/
MSG6: .ASCIZ /DRIVE #/
  
```

.EVEN

```

1
2
3
4
5
6
7
8
9
10
11
12 030032 000000
13 030034
22 030034
23
30
31 030034
32
38 030200 000000
   030202 000000
   030204 000000
   030206 000000
   030210 000000
   030212 000000
42
48 030214 000000
   030216 000000
   030220 000000
   030222 000000
   030224 000000
   030226 000000
52
58 030230 000000
   030232 000000
   030234 000000
   030236 000000
   030240 000000
   030242 000000
62
63 030244
64 030244
65 030250
66 031250
67 032246
68
69 032250
70 032250
71 033400
72 034530
73 035656
74 035656
75
76 035660
77 117656
78

.SBTTL SOFTWARE PARAMETER CODING SECTION

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

LSSOFT: .WORD L10020-LSSOFT/2
L10020: .EVEN

$PATCH: .BLKW 50. ;PROGRAM PATCH AREA (50. WORDS)

TDWRD1: .WORD 0 ;TD WORD #1
TDWRD2: .WORD 0 ;TD WORD #2
TDWRD3: .WORD 0 ;TD WORD #3
TDWRD4: .WORD 0 ;TD WORD #4
TDWRD5: .WORD 0 ;TD WORD #5
TDWRD6: .WORD 0 ;TD WORD #6

TDCPY1: .WORD 0 ;COPY OF TD WORD #1
TDCPY2: .WORD 0 ;COPY OF TD WORD #2
TDCPY3: .WORD 0 ;COPY OF TD WORD #3
TDCPY4: .WORD 0 ;COPY OF TD WORD #4
TDCPY5: .WORD 0 ;COPY OF TD WORD #5
TDCPY6: .WORD 0 ;COPY OF TD WORD #6

TDTMP1: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #1
TDTMP2: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #2
TDTMP3: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #3
TDTMP4: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #4
TDTMP5: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #5
TDTMP6: .WORD 0 ;TEMPORARY STORAGE FOR TD WORD #6

BEGBUF = . ;BEGINNING OF BUFFER SECTION
TMPBSF: .BLKW 2 ;GENERAL BSF USE
MFGBUF: .BLKW 256. ;MFG BUFFER FOR DEC 144 FILE
USRBUF: .BLKW 256. ;USR BUFFER FOR DEC 144 FILE
ENDUSR = -2 ;END OF USR BUFFER

DEFBUF = . ;BEGINNING OF DEFECT BUFFER (SCAN MODE)
HDRBLK: .BLKW 6*50. ;HEADER BLOCK (6 WRDS X 50. SECTORS)
TDSBLK: .BLKW 6*50. ;TRACK DESCRIPTOR BLOCK
LINBUF: .BLKW 6*50. ;BEGINNING OF LINEAR BUFFER
ENDLIN = -2 ;END OF LINEAR BUFFER
ENDDEF = ENDLIN ;END OF DEFECT BUFFER

DBUFF: .BLKW 256.*50. ;DATA BUFFER (256. WRDS X 50. SECTORS)
ENDBUF = -2 ;END OF BUFFER SECTION
    
```


85
117660 117700
117662 000006
117664

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

1
14
16 117664 000000
117666 000004
117670
17 117670 176700
18 117672 000254
19 117674 000240
20 117676 000000
21 117700
23 000001

L10021: .WORD 0
.WORD L10023-./2-1
.WORD 17670C
.WORD 254
.WORD 240
.WORD 0
L10023:
.END

ADR = 000020 G	CMOD = 100000 G	C\$REVI= 000003	ECI = 004000 G	FINDWR 022634
ANDHDR 005713 G	CNTDEF 017750	C\$RFLA= 000021	EF.CON= 000036 G	FINSCR 024104
AOE = 001000 G	COMPRS 024054	C\$RPT = 000025	EF.NEW= 000035 G	FIRPAS 002235 G
ASSEMB= 000010	CONTIN 002152 G	C\$SEFG= 000046	EF.PWR= 000034 G	FMTRK = 000063 G
ATA = 100000 G	CONTLT 002346 G	C\$SPRI= 000041	EF.RES= 000037 G	FMT16 = 010000 G
ATO = 000001 G	CPE = 040000 G	C\$SVEC= 000037	EF.STA= 000040 G	FORMAT 005510 G
AT1 = 000002 G	CPU = 020000 G	C\$TPRI= 000013	EM1 006527 G	FORMT 002162 G
AT2 = 000004 G	CR = 000015 G	DASCAN 022220	EM10 006756 G	FRMTT 002645 G
AT3 = 000010 G	CRE144 017664	DATA = 000402 G	EM11 007004 G	FRMT00 002650 G
AT4 = 000020 G	CRLF 002642 G	DBUFF = 035660 G	EM12 007030 G	FRMT01 002741 G
AT5 = 000040 G	CURENT 005640 G	DCK = 100000 G	EM13 007067 G	FRMT02 003021 G
AT6 = 000100 G	C\$AU = 000052	DCU = 000040 G	EM14 007112 G	FRMT03 003112 G
AT7 = 000200 G	C\$AUTO= 000061	DEFBUF= 032250	EM15 007140 G	FRMT04 003172 G
A16 = 000400 G	C\$BRK = 000022	DEF CNT 002246 G	EM16 007141 G	FRMT05 003264 G
A17 = 001000 G	C\$BSEG= 000004	DEF SEC 002372 G	EM17 007216 G	FRMT06 003346 G
BADDR 002230 G	C\$BSUB= 000002	DEF TRK 002374 G	EM2 006554 G	FRMT07 003366 G
BAI = 000010 G	C\$CEFG= 000045	DEF1 002274 G	EM20 007263 G	FRMT10 003405 G
BEGBUF = 030244	C\$CLCK= 000062	DEF2 002276 G	EM21 007341 G	FRMT11 003423 G
BELL = 000007 G	C\$CLEA= 000012	DEF3 002300 G	EM22 007364 G	FRMT12 003516 G
BGNPAT= 002142 G	C\$CLOS= 000035	DEF4 002302 G	EM3 006600 G	FRMT13 003564 G
BIT0 = 000001 G	C\$CLP1= 000006	DELTA 002554 G	EM4 006624 G	FRMT14 003624 G
BIT00 = 000001 G	C\$CVEC= 000036	DESCYL 002304 G	EM5 006664 G	FRMT15 003635 G
BIT01 = 000002 G	C\$DCLN= 000044	DESTD 025112	EM6 006707 G	FRMT16 003645 G
BIT02 = 000004 G	C\$DODU= 000051	DESTRK 002244 G	EM7 006725 G	FRMT17 003740 G
BIT03 = 000010 G	C\$DRPT= 000024	DFPTBL 002130 G	ENDBUF= 117656	FRMT20 004020 G
BIT04 = 000020 G	C\$DU = 000053	DIAG = 000035 G	ENDCYL 002216 G	FRMT21 004054 G
BIT05 = 000040 G	C\$EDIT= 000003	DIAGMC= 000000	ENDDEF= 035656	FRMT22 004103 G
BIT06 = 000100 G	C\$ERDF= 000055	DLT = 100000 G	ENDLIN= 035656	FRMT23 004125 G
BIT07 = 000200 G	C\$ERHR= 000056	DMD = 100000 G	ENDPAT= 002152 G	FRMT24 004146 G
BIT08 = 000400 G	C\$ERRO= 000060	DMPREG 010556	ENDPTR 002224 G	FRMT30 004216 G
BIT09 = 001000 G	C\$ERSF= 000054	DPE = 000010 G	ENDTAB 002226 G	FRMT31 004244 G
BIT1 = 000002 G	C\$ERSO= 000057	DPR = 000400 G	ENDTRK 002220 G	FRMT32 004266 G
BIT10 = 002000 G	C\$ESCA= 000010	DRIVER 010166	ENDUSR= 032246	FRMT33 004364 G
BIT11 = 004000 G	C\$ESEG= 000005	DRQ = 004000 G	ENWTD 002172 G	FRMT35 004421 G
BIT12 = 010000 G	C\$ESUB= 000003	DRT0 = 000001 G	ERR = 040000 G	FRMT36 004530 G
BIT13 = 020000 G	C\$FTST= 000001	DRT1 = 000002 G	ERRFNC 002254 G	FRMT37 004600 G
BIT14 = 040000 G	C\$EXIT= 000032	DRT2 = 000004 G	ERRMAX 002222 G	FRMT50 004647 G
BIT15 = 100000 G	C\$GETB= 000026	DRT3 = 000010 G	ERRMSK 002264 G	FRMT60 005005 G
BIT2 = 000004 G	C\$GETW= 000027	DRT4 = 000020 G	ERRORS 012752	FRMT61 005064 G
BIT3 = 000010 G	C\$GMAN= 000043	DRT5 = 000040 G	ERRTTL 002262 G	FRMT62 005156 G
BIT4 = 000020 G	C\$GPHR= 000042	DRT6 = 000100 G	ERRVEC= 000004	FRMT63 005237 G
BIT5 = 000040 G	C\$GPLO= 000030	DRT7 = 000200 G	ERRO 007406 G	FRMT64 005327 G
BIT6 = 000100 G	C\$GPRI= 000040	DRT8 = 000400 G	ERR1 007440 G	FUNCTN 002370 G
BIT7 = 000200 G	C\$INIT= 000011	DRVCLR= 000011 G	ERR2 007562 G	F\$AU = 000015
BIT8 = 000400 G	C\$INLP= 000020	DRVINI 013460	ERR3 007704 G	F\$AUTO= 000020
BIT9 = 001000 G	C\$MANI= 000050	DRVNO 002420 G	ERR5 007732 G	F\$BGN = 000040
BOE = 000400 G	C\$MEM = 000031	DRVPAR 002156 G	ERTTL 002260 G	F\$CLEA= 000007
BSE = 100000 G	C\$MSG = 000023	DRVRDY 005567 G	EVL = 000004 G	F\$DU = 000016
BUF INI 014134	C\$OPEN= 000034	DRVSN 002422 G	EWN = 000002 G	F\$END = 000041
CALCTD 015416	C\$PNTB= 000014	DRY = 000200 G	EXPCTD 002270 G	F\$HARD= 000004
CEMODE 002234 G	C\$PNTF= 000017	DS = 000154 G	EXPTD 005612 G	F\$HW = 000013
CHABUF 011264	C\$PNTS= 000016	DSE = 020000 G	ESEND = 002100	F\$INIT= 000006
CHANGE 006000 G	C\$PNTX= 000015	DTE = 010000 G	E\$LOAD= 000035	F\$JMP = 000050
CHEKTD 023424	C\$QIO = 000377	DVA = 004000 G	FER = 000020 G	F\$MOD = 000000
CHKSEG 016400	C\$RDBU= 000007	DVC = 000200 G	FETWO 020120	F\$MSG = 000011
CLR = 000040 G	C\$REFG= 000047	ECCWRD 002364 G	FINDBA 023130	F\$PROT= 000021
CMDQUE 002564 G	C\$RESE= 000033	ECH = 000100 G	FINDSE 022712	F\$PWR = 000017

F\$RPT = 000012	INPUTD 011370	L\$DU 026644 G	MAXCYL 002206 G	NORUN 014122
F\$SEG = 000003	INSERT 014522	L\$DUT 002072 G	MAXSEG 002400 G	NOWRCK 002164 G
F\$SOFT = 000005	INTEN = 000100 G	L\$DVTY 002600 G	MAXTRK 002212 G	NULL = 140000 G
F\$SRV = 000010	INTLEV 002237 G	L\$SEF 002052 G	MAXVAL 023736	OFFDIR = 000200 G
F\$SUB = 000002	IR = 000100 G	L\$SENV 002044 G	MCPE = 020000 G	OM = 000001 G
F\$SW = 000014	ISR = 000100 G	L\$SETP 002102 G	MERGE 016610	ONEFIL = 000001
F\$TEST = 000001	IXE = 004000 G	L\$SEXP1 002046 G	MESG1 027762	OPI = 020000 G
F1 = 000002 G	IXU = 000100 G	L\$SEXP4 002064 G	MESG13 006115 G	OPTION 002160 G
F2 = 000004 G	ISAU = 000041	L\$SEXP5 002066 G	MESG14 006125 G	OR = 000200 G
F3 = 000010 G	ISAUTO = 000041	L\$SHARD 027716 G	MESG15 006135 G	OSAPTS = 000000
F4 = 000020 G	ISCLN = 000041	L\$SHIME 002120 G	MESG16 006145 G	OSAU = 000000
F5 = 000040 G	ISDU = 000041	L\$SHPCP 002016 G	MESG17 006310 G	OSBGNR = 000000
GAPVAL 023546	ISHRD = 000041	L\$SHPTP 002022 G	MESG18 006037 G	OSBGNS = 000000
GETNEX 014316	ISINIT = 000041	L\$SHW 002130 G	MESG19 006403 G	OSDU = 000000
GETPAT 023054	ISMOD = 000041	L\$SICP 002104 G	MESG22 006334 G	OSERRT = 000000
GO = 000001 G	ISMSG = 000041	L\$SLADP 002026 G	MESG23 006462 G	OSGNSW = 000000
G\$CNTD = 000200	ISPROT = 000040	L\$SLAST 117664 G	MESG25 006155 G	OSPOIN = 000001
G\$DELM = 000372	ISPTAB = 000041	L\$LOAD 002100 G	MESG38 012214	OSSETU = 000001
G\$DISP = 000003	ISPRW = 000041	L\$LUN 002074 G	MESG39 012235	PAR = 000010 G
G\$EXCP = 000400	ISRPT = 000041	L\$MREV 002050 G	MESG4 027775	PARTBL 002322 G
G\$HILI = 000002	ISSEG = 000041	L\$MREV 002050 G	MESG40 012253	PAT = 000020 G
G\$LOLI = 000001	ISSETU = 000041	L\$NAME 002000 G	MESG41 012272	PATRN 002312 G
G\$NO = 000000	ISSFT = 000041	L\$PRIO 002042 G	MESG42 012313	PGE = 100000 G
G\$OFFS = 000400	ISSRV = 000041	L\$PROT 025332 G	MESG43 012332	PGM = 001000 G
G\$OFFSI = 000376	ISSUB = 000041	L\$PRT 002112 G	MESG44 012344	PHF = 000400 G
G\$PRMA = 000001	ISTST = 000041	L\$REPP 002062 G	MESG45 012356	PIP = 020000 G
G\$PRMD = 000002	J\$JMP = 000167	L\$REV 002010 G	MESG46 012370	PLTRK 002307 G
G\$PRML = 000000	LASLOC 002232 G	L\$RPT 025324 G	MESG47 012402	PNT = 001000 G
G\$RADA = 000140	LBC = 002000 G	L\$SOFT 030034 G	MESG5 030011	PRI = 002000 G
G\$RADB = 000000	LBT = 002000 G	L\$SPC 002056 G	MESG6 030022	PRI00 = 000000 G
G\$RADD = 000040	LCE = 001000 G	L\$SPCP 002020 G	MESG7 006067 G	PRI01 = 000040 G
G\$RADL = 000120	LF = 000012 G	L\$SPTP 002024 G	MFGBUF 030250 G	PRI02 = 000100 G
G\$RADO = 000020	LINBUF 034530 G	L\$STA 002030 G	MINCYL 002204 G	PRI03 = 000140 G
G\$XFER = 000004	LISHDR 002202 G	L\$SW 002142 G	MINSEG 002376 G	PRI04 = 000200 G
G\$YES = 000010	LIST 002200 G	L\$STEST 002114 G	MINTRK 002210 G	PRI05 = 000240 G
G1 = 000100 G	LISTDS 020742	L\$TIML 002014 G	MODBY 002176 G	PRI06 = 000300 G
G2 = 000045 G	LOADTD 024744	L\$UNIT 002012 G	MODTD 002174 G	PRI07 = 000340 G
G3 = 000040 G	LOCYL 002360 G	L10000 002140	MOH = 020000 G	PSEL = 002000 G
G4 = 000045 G	LODBSF 011122	L10001 002142	MOL = 010000 G	RCVED 005625 G
HCE = 000200 G	LOE = 040000 G	L10002 007436	MOLINE 005537 G	RDBSF 017164
HCI = 002000 G	LOT = 000010 G	L10003 007560	MORETD 002402 G	RDDAT = 000071 G
HCRC = 000400 G	L\$ACP 002110 G	L10004 007702	MPE = 000400 G	RDHD = 000073 G
HDR = 000010 G	L\$APT 002036 G	L10005 007730	MSEG = 000022 G	RDTD = 000075 G
HDRBLK 032250 G	L\$AU 026652 G	L10006 010012	MSRDTD 005425 G	RDY = 000200 G
HDRO = 000010 G	L\$AUT 002070 G	L10007 025330	MSWRTD 005414 G	READIN = 000021 G
HDSCAN 021726	L\$AUTO 026334 G	L10008 026332	MTD = 040000 G	READTD 020466
HELP = 000000	L\$CCP 002106 G	L10009 026334	MXF = 001000 G	READY 010156
HICYL 002356 G	L\$CLEA 026336 G	L10010 026642	NBA = 100000 G	RECAL = 000007 G
HIVALU 023474	L\$CO 002032 G	L10011 026650	NED = 010000 G	RECALI 005502 G
HOE = 100000 G	L\$DEPO 002011 G	L10012 026656	NEGWRD 002362 G	RCVED 002272 G
IAE = 002000 G	L\$DESC 002606 G	L10013 027712	NEM = 004000 G	REG 002500 G
IBE = 010000 G	L\$DESP 002076 G	L10014 027762	NEWCNT 002250 G	RELSE = 000013 G
IDU = 000040 G	L\$DEVP 002060 G	L10015 030034	NEWDAT 025164	RELTIV 014372
IER = 020000 G	L\$DISP 002124 G	L10016 117670	NEWLST 021252	RESET 012540
ILEV = 000004 G	L\$DLY 002116 G	L10017 117700	NOGOOD 014126	RESTOR 025234
ILF = 000001 G	L\$DTP 002040 G	L10018 013302	NOOP = 000001 G	RESTTD 025054
ILR = 000002 G	L\$DTYP 002034 G	MAPSEC 014452	NOROOM 002241 G	RETRY 012446

RHDATA	005523	G	SCANR	002170	G	TDCPY5	030224	G	T\$HILI=	000007	T1	026660	G		
RHEXT	002414	G	SCF	= 000002	G	TDCPY6	030226	G	T\$LAST=	000001	UAM	= 000200	G		
RHTYPE	002416	G	SC1	= 000100	G	TDDEF	005656	G	T\$LOLI=	000000	UNIT	= 002404	G		
RMR	= 000004	G	SC16	= 002000	G	TDDUMP	015014	G	T\$LSYM=	010000	UNS	= 040000	G		
RPADR	002406	G	SC2	= 000200	G	TDLMTS	002336	G	T\$LTNO=	000001	UPDSCR	023220			
RPAS	002442	G	SC32	= 004000	G	TDSBLK	033400	G	T\$NEST=	177777	UPE	= 020000	G		
RPBA	002430	G	SC4	= 000400	G	TDSCAN	021572	G	T\$NS0 =	000000	USRBUF	031250	G		
RPBAE	002474	G	SC64	= 010000	G	TDTMP1	030230	G	T\$NS1 =	000005	US1	= 000001	G		
RPCC	002462	G	SC8	= 001000	G	TDTMP2	030232	G	T\$PCNT=	000000	US2	= 000002	G		
RPCS1	002424	G	SDF	= 000020	G	TDTMP3	030234	G	T\$PTAB=	010022	US4	= 000004	G		
RPCS2	002434	G	SEARCH=	000031	G	TDTMP4	030236	G	T\$PTHV=	000001	VERIFY	017064			
RPCS3	002476	G	SECADD	002366	G	TDTMP5	030240	G	T\$PTNU=	000001	VRIFY	= 002166	G		
RPDA	002432	G	SEC50	002306	G	TDTMP6	030242	G	T\$SAVL=	177777	VV	= 000100	G		
RPDB	002446	G	SEEK	= 000005	G	TDVALU	002316	G	T\$SEGL=	177777	WAIT	012720			
RPDC	002460	G	SEIZE	012602		TDWRD1	030200	G	T\$SIZE=	000006	WARN	= 006016	G		
RPDS	002436	G	SESORT	015316		TDWRD2	030202	G	T\$SUBN=	000000	WCE	= 040000	G		
RPDT	002452	G	SFPTBL	002142	G	TDWRD3	030204	G	T\$TAGL=	177777	WCF	= 000040	G		
RPEC1	002470	G	SIZE70	010014		TDWRD4	030206	G	T\$TAGN=	010024	WCKD	= 000051	G		
RPEC2	002472	G	SKI	= 040000	G	TDWRD5	030210	G	T\$TEMP=	000000	WCKDAT	005446	G		
RPER1	002440	G	SOF SW	002256	G	TDWRD6	030212	G	T\$TEST=	000001	WCKHD	= 000053	G		
RPER2	002464	G	SPIRAL	017720		TEMPA	002214	G	T\$TSTM=	177777	WCKHDR	005463	G		
RPER3	002466	G	SUPRSS	002242	G	TEMP1	002352	G	T\$TSTS=	000001	WLE	= 004000	G		
RPLA	002444	G	SVCGBL=	000000		TEMP2	002354	G	T\$\$AU =	010015	WLOCK	005553	G		
RPMR1	002450	G	SVCINS=	000000		TEXT	002344	G	T\$\$AUT=	010012	WOR	= 001000	G		
RPOF	002456	G	SVC SUB=	000000		TKWCNT	002350	G	T\$\$CLE=	010013	WRDAT	005435	G		
RPSN	002454	G	SVCTAG=	000000		TMDLTA	016250		T\$\$DAT=	010023	WRITCK	011212			
RPVEC	002410	G	SVCTST=	000000		TMPBSF	030244	G	T\$\$DU =	010014	WRITTD	020222			
RPWC	002426	G	S\$LSYM=	010000		TOLER	022744		T\$\$HAR=	010017	WRL	= 004000	G		
RTYCNT	002266	G	TABELD	014152	G	TOLRAN	023336		T\$\$HW =	010000	WRTBSF	017436			
RWU1	= 002000	G	TAP	= 040000	G	TRE	= 040000	G	T\$\$INI=	010011	WRTDAT=	000061	G		
RWU2	= 004000	G	TBLPTR	002252	G	TYPLOC	010472		T\$\$MSG=	010006	WRTFE2	002154	G		
RWU3	= 010000	G	TD	= 000010	G	T\$ARGC=	000002		T\$\$PC =	000001	WRITD	= 000065	G		
SAMSEC	002240	G	TDBCNT	002314	G	T\$CODE=	003032		T\$\$PRO=	010010	WRU	= 000400	G		
SAVETD	025016		TDCASE	016252		T\$ERRN=	000002	T\$\$PTA=	010022	X\$ALWA=	000000				
SAVRPR	017012		TDCPY1	030214	G	T\$EXCP=	000000	T\$\$RPT=	010007	X\$FALS=	000040				
SBE	= 000004	G	TDCPY2	030216	G	T\$FLAG=	000040	T\$\$SOF=	010020	X\$OFFS=	000400				
SC	= 100000	G	TDCPY3	030220	G	T\$FREE=	117700	T\$\$SW =	010001	X\$TRUE=	000020				
SCAN	027156		TDCPY4	030222	G	T\$GMAN=	000000	T\$\$TES=	010016	\$PATCH	030034	G			
SCANIT	002236	G													

. ABS. 117700 000
 000000 001
 ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 29696 WORDS (116 PAGES)
 DYNAMIC MEMORY AVAILABLE FOR 70 PAGES
 (CZRJKA.BIC,CZRJKA/C=[20,0]SVC34R.MLB,[20,12]CZRJKA.DOC,CZRJKA.HIS,CZRJKA

IER	11-57#						
ILEV	12-63#	25-20					
ILF	12-78#						
ILR	12-79#						
INPUTD	21-13#	65-104					
INSERT	21-75	29-16#	48-50	49-77	53-48		
INTEN	12-5#						
INTLEV	13-52#	20-15	25-22*	28-41	35-17	37-18	60-113
IR	12-31#						
ISR	11-57#						
IXE	11-57#						
IXU	12-174#						
J\$JMP	7-143#	58-61	63-19	64-20			
L\$ACP	7-197#						
L\$APT	7-197#						
L\$AU	64-9#						
L\$AUT	7-197#						
L\$AUTO	7-197	61-10#					
L\$CCP	7-197#						
L\$CLEA	7-197	62-8#					
L\$CO	7-197#						
L\$DEPO	7-197#						
L\$DESC	7-197	14-27#					
L\$DESP	7-197#						
L\$DEVP	7-197#						
L\$DISP	7-197	8-8#					
L\$DLY	7-197#	22-104					
L\$DTP	7-197#						
L\$DTYP	7-197#						
L\$DU	63-8#						
L\$DUT	7-197#						
L\$DVTY	7-197	14-17#					
L\$EF	7-197#						
L\$ENVI	7-197#						
L\$ETP	7-197#						
L\$EXP1	7-197#						
L\$EXP4	7-197#						
L\$EXP5	7-197#						
L\$HARD	7-197	68-53	68-53#				
L\$HIME	7-197#						
L\$HPCP	7-197#						
L\$HPTP	7-197#						
L\$HW	7-197	9-10	9-10#				
L\$ICP	7-197#						
L\$INIT	7-197	60-8#					
L\$LADP	7-197#						
L\$LAST	7-197	69-85#	70-22				
L\$LOAD	7-197#						
L\$LUN	7-197#						
L\$MREV	7-197#						
L\$NAME	7-197#						
L\$PRIO	7-197#						
L\$PROT	7-197	59-8#					
L\$PRT	7-197#						
L\$REPP	7-197#						
L\$REV	7-197#						

RPADR	13-126#	26-39*												
RPAS	13-140#	22-92*												
RPBA	13-135#	17-40*												
RPBAE	13-153#													
RPCC	13-148#													
RPCS1	13-133#	17-45*	17-46	17-53	17-62	22-53*	25-72*	26-19	36-15					
RPCS2	13-137#	17-69	22-73*	22-74*	25-14	62-11*	62-12*							
RPCS3	13-154#													
RPDA	13-136#	17-39*												
RPDB	13-142#													
RPDC	13-147#	17-38*												
RPDS	13-138#	17-25	17-31	17-60	17-67	22-75	22-77*	22-80	25-13	25-20	25-37	25-47	25-61	25-73
RPDT	13-144#	25-28												
RPEC1	13-151#													
RPEC2	13-152#													
RPER1	13-139#													
RPER2	13-149#													
RPER3	13-150#													
RPLA	13-141#													
RPMR1	13-143#	17-41*	17-44*	22-57*										
RPOF	13-146#	20-21*	22-54*	24-37*	43-36*	43-51*	44-29*	48-19*						
RPSN	13-145#													
RPVEC	13-127#	26-40*	26-41*											
RPWC	13-134#	17-37*												
RTYCNT	13-66#	22-27*	22-29	22-34*	24-42	25-54	25-68	25-79	25-87	38-31	39-29	39-41	43-42	44-37
	48-25	49-28	49-43	49-58*	65-70*									
RWU1	12-188#													
RWU2	12-189#													
RWU3	12-190#													
S&LSYM	7-143#	9-25#	10-20#	15-14#	15-25#	15-34#	15-40#	15-49#	21-15	21-15	21-15	21-15#	21-17	21-17
	21-17	21-17#	21-19	21-19	21-19	21-19#	21-45	21-45	21-45	21-45#	21-49	21-49	21-49	21-49#
	21-53	21-53	21-53	21-53#	21-57	21-57	21-57	21-57#	21-64	21-64	21-64	21-64#	21-66	21-66
	21-66	21-66#	21-74	21-74	21-74	21-74#	21-109	21-109	21-109	21-109#	58-76#	60-37	60-37	60-37
	60-37#	60-48	60-48	60-48	60-48#	60-60	60-60	60-60	60-60#	60-64	60-64	60-64	60-64#	60-66
	60-66	60-66	60-66#	60-68	60-68	60-68	60-68#	60-70	60-70	60-70	60-70#	60-75	60-75	60-75
	60-75#	60-85	60-85	60-85	60-85#	60-90	60-90	60-90	60-90#	60-217#	61-18#	62-62#	63-34#	64-35#
	66-152#	68-62#	69-22#											
SAMSEC	13-53#	32-54*	32-84	32-101	32-142*									
SAVETD	21-88	55-39#	66-92											
SAVRPR	22-24	22-88	36-14#											
SBE	12-170#													
SC	12-12#													
SC1	12-136#													
SC16	12-140#													
SC2	12-137#													
SC32	12-141#													
SC4	12-138#													
SC64	12-142#													
SC8	12-139#													
SCAN	66-5#													
SCANIT	13-51#	46-18*	46-20	66-5*										
SCANR	13-25#	29-42	44-57	60-106*	60-133*	65-86								
SCF	12-169#													
SDF	12-172#													
SEARCH	12-215#													
SEC50	13-75#	20-43	35-50	35-55	37-22	39-44	50-47							

	13-117#	21-64	28-14	28-20*	28-26*	28-45	28-51*	29-40	48-54	49-81	50-44*	50-45*	50-47*	50-50*
SECADD	13-117#	21-64	28-14	28-20*	28-26*	28-45	28-51*	29-40	48-54	49-81	50-44*	50-45*	50-47*	50-50*
	53-21*	53-24*	53-53											
SEEK	12-206#													
SEIZE	22-52	22-73#	25-10											
SESORT	31-7#	65-133	66-130											
SFPTBL	10-10#													
SIZE70	16-11#	26-29												
SKI	12-180#													
SOF SW	13-62#	25-92*	65-76											
SPIRAL	40-42#	65-130	66-48											
SUPRSS	13-56#	21-94*	21-98*	21-106*	22-86	22-87*	22-90*	23-16	38-17*	38-57*	47-31*	49-32*	49-49*	51-38*
	60-103*	66-18*	66-42*											
SVCGBL	7-143#	7-152#	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	10-10	10-10	10-10	10-10	10-10	14-17	14-17	14-17	14-27	14-27	14-27	15-9	15-9	15-9
	15-16	15-16	15-16	15-27	15-27	15-27	15-36	15-36	15-36	15-42	15-42	15-42	58-47	58-47
	58-47	59-8	59-8	59-8	60-8	60-8	60-8	61-10	61-10	61-10	62-8	62-8	62-8	63-8
	63-8	63-8	64-9	64-9	64-9	68-53	68-53	68-53	69-12	69-12	69-12	69-85	69-85	69-85
	69-85#													
SVCINS	7-143#	7-149#	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	8-8	8-8	9-10	9-10	10-10	10-10	14-17	14-17	14-17	14-17	14-27	14-27	14-27	14-27
	15-13	15-13	15-13	15-13	15-13	15-13	15-13	15-13	15-13	15-13	15-14	15-14	15-19	15-19
	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19	15-19
	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21	15-21
	15-21	15-21	15-24	15-24	15-24	15-24	15-24	15-24	15-24	15-24	15-24	15-24	15-25	15-25
	15-30	15-30	15-30	15-30	15-30	15-30	15-30	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	15-31	15-31	15-31	15-31	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	15-33	15-33	15-33	15-33	15-33
	15-33	15-33	15-34	15-34	15-39	15-39	15-39	15-39	15-39	15-39	15-39	15-39	15-39	15-39
	15-40	15-40	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-45
	15-45	15-45	15-48	15-48	15-48	15-48	15-48	15-48	15-48	15-48	15-48	15-48	15-49	15-49
	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-16	18-20	18-20
	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-20	18-37	18-37
	18-37	18-37	18-37	18-37	18-37	18-37	18-37	18-37	18-38	18-38	18-38	18-38	18-38	18-38
	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38
	18-38	18-38	18-38	18-38	18-40	18-40	18-40	18-40	18-40	18-40	18-40	18-40	18-40	18-40
	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41
	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-43	18-43	18-43	18-43
	18-43	18-43	18-43	18-43	18-43	18-43	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44
	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44
	18-44	18-44	18-49	18-49	18-49	18-49	18-49	18-49	18-49	18-49	18-49	18-49	18-50	18-50
	18-50	18-50	18-50	18-50	18-50	18-50	18-50	18-50	18-50	18-50	18-50	18-50	19-28	19-28
	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	21-15	21-15	21-15	21-15

	60-85	60-85	60-85	60-85	60-85	60-85	60-85	60-85	60-85	60-85	60-90	60-90	60-90	60-90
	60-90	60-90	60-90	60-90	60-90	60-90	60-90	60-90	60-90	60-96	60-96	60-97	60-97	60-97
	60-99	60-99	60-99	60-99	60-202	60-202	60-202	60-202	60-202	60-217	60-217	61-18	61-18	62-10
	62-10	62-10	62-14	62-14	62-14	62-14	62-14	62-14	62-14	62-14	62-14	62-14	62-14	62-17
	62-17	62-17	62-21	62-21	62-21	62-21	62-21	62-21	62-21	62-21	62-21	62-21	62-21	62-21
	62-21	62-21	62-24	62-24	62-24	62-24	62-24	62-24	62-24	62-24	62-24	62-24	62-24	62-24
	62-24	62-24	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27
	62-27	62-27	62-32	62-32	62-32	62-32	62-32	62-32	62-32	62-32	62-32	62-32	62-32	62-32
	62-32	62-32	62-34	62-34	62-34	62-34	62-34	62-34	62-34	62-34	62-34	62-34	62-34	62-34
	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36	62-36
	62-47	62-47	62-47	62-47	62-62	62-62	62-62	62-62	62-62	62-62	62-62	62-62	62-62	62-62
	64-20	64-20	64-35	64-35	65-78	65-78	65-78	65-78	65-78	65-83	65-83	65-83	65-83	65-92
	65-102	65-102	65-103	65-103	65-135	65-135	65-135	65-135	65-135	66-15	66-15	66-15	66-15	66-15
	66-15	66-15	66-15	66-15	66-15	66-15	66-15	66-15	66-15	66-152	66-152	66-152	66-152	66-152
	68-55	68-55	68-55	68-55	68-55	68-55	68-55	68-55	68-55	68-57	68-57	68-57	68-57	68-57
	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59	68-59
	68-61	68-61	68-61	68-61	68-61	68-61	68-61	68-61	68-61	68-62	68-62	68-62	68-62	68-62
	69-85	69-85	69-85	69-85	70-16	70-16	70-16	70-16	70-16	69-12	69-12	69-12	69-12	69-12
SVCSUB	7-143#	7-151#												
SVCTAG	7-143#	7-153#	9-25	9-25	9-25	10-20	10-20	10-20	15-14	15-14	15-14	15-25	15-25	15-25
	15-34	15-34	15-34	15-40	15-40	15-40	15-49	15-49	15-49	21-15	21-15	21-15	21-15	21-17
	21-17	21-19	21-19	21-19	21-45	21-45	21-45	21-45	21-49	21-49	21-49	21-53	21-53	21-53
	21-57	21-57	21-64	21-64	21-64	21-66	21-66	21-66	21-74	21-74	21-74	21-109	21-109	21-109
	58-76	58-76	58-76	60-37	60-37	60-37	60-48	60-48	60-48	60-48	60-48	60-60	60-60	60-64
	60-64	60-66	60-66	60-66	60-68	60-68	60-68	60-68	60-70	60-70	60-70	60-75	60-75	60-75
	60-85	60-85	60-90	60-90	60-90	60-217	60-217	60-217	60-217	60-217	60-217	60-60	60-60	60-64
	63-34	63-34	63-34	64-35	64-35	64-35	66-152	66-152	66-152	66-152	66-152	68-62	68-62	62-62
	69-22	70-16	70-16	70-16	70-21	70-21	70-21	70-21	70-21	61-18	61-18	61-18	62-62	62-62
SVCTST	7-143#	7-150#	65-68	65-68	65-68									
TSSAU	64-9#	64-20	64-35											
TSSAUT	61-10#	61-18												
TSSCLE	62-8#	62-17	62-47	62-62										
TSSDAT	70-16	70-16#	70-21											
TSSDU	63-8#	63-19	63-34											
TSSHAR	68-53	68-53#	68-62											
TSSHW	9-10	9-10#	9-25											
TSSINI	60-8#	60-16	60-97	60-202	60-217									
TSSMSG	15-9#	15-14	15-16#	15-25	15-27#	15-34	15-36#	15-40	15-42#	15-49				
TSSPC	70-15#	70-22												
TSSPRO	59-8#													
TSSPTA	70-15#	70-16	70-16#											
TSSRPT	58-47#	58-61	58-76											
TSSSOF	69-12	69-12#	69-22											
TSSSW	10-10	10-10#	10-20											
TSSTES	65-68#	65-78	65-83	65-135	66-152									
T\$ARGC	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197#	7-197#	7-197#
	7-197#	7-197#	7-197#	15-13	15-13	15-13#	15-19	15-19	15-19	15-19	15-19	15-19#	15-19#	15-19#
	15-19#	15-21	15-21	15-21	15-21	15-21#	15-21#	15-21#	15-21#	15-21#	15-21#	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#	15-31#	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#	15-45#	15-45#	15-48	15-48	15-48#	18-16	18-16	18-16	18-16#	18-16#	18-20	18-20
	18-20	18-20	18-20#	18-20#	18-20#	18-37	18-37	18-37#	18-38	18-38	18-38#	18-38#	18-38	18-38
	18-38	18-38	18-38	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-40	18-40#
	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41#	18-41#	18-41#	18-41#	18-41#
	18-41#	18-41#	18-41#	18-43	18-43	18-43#	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44
	18-44	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-49	18-49	18-49#	18-50

	18-50	18-50	18-50#	18-50#	18-50#	19-28	19-28	19-28	19-28#	19-28#	21-43	21-43	21-43#	25-12
	25-12	25-12	25-12#	25-12#	25-24	25-24	25-24#	25-27	25-27#	25-27#	29-46	29-46	29-46#	29-46#
	29-46#	30-20	30-20	30-20#	30-20#	30-20#	30-22	30-22	30-22#	30-33	30-33	30-33	30-33#	30-33
	30-33	30-33	30-33#	30-33#	30-33#	30-33#	30-33#	30-33#	30-39	30-39	30-39	30-39#	30-39#	30-42
	30-42	30-42	30-42#	30-42#	30-46	30-46	30-46#	38-40	38-40	38-40	38-40#	38-40#	41-38	41-38
	41-38	41-38	41-38#	41-38#	41-38#	41-38#	41-38#	45-43	45-43	45-43	45-43	45-43#	45-43#	45-43#
	45-48	45-48	45-48#	45-48#	45-48#	45-51	45-51	45-51	45-51#	45-51#	45-55	45-55	45-55#	46-23
	46-23	46-23	46-23#	46-23#	46-26	46-26	46-26#	46-42	46-42	46-42	46-42	46-42	46-42	46-42
	46-42#	46-42#	46-42#	46-42#	46-42#	46-46	46-46#	46-46	46-46#	46-46#	46-46#	46-49	46-49	46-49
	46-49#	46-49#	46-54	46-54	46-54#	60-35	60-35	60-35#	62-14	62-14	62-14#	62-21	62-21	62-21
	62-21	62-21#	62-21#	62-21#	62-24	62-24	62-24#	62-24	62-24#	62-24#	62-24#	62-27	62-27	62-27
	62-27	62-27#	62-27#	62-27#	62-32	62-32	62-32#	62-32	62-32#	62-32#	62-32#	62-34	62-34	62-34
	62-34#	62-34#	62-36	62-36	62-36#	62-36	62-36#	62-36#	62-36#	66-15	66-15	66-15	66-15#	66-15#
T\$CODE	21-15	21-15	21-15#	21-15#	21-15#	21-15#	21-17	21-17	21-17#	21-17#	21-17#	21-17#	21-19	21-19
	21-19	21-19#	21-19#	21-19#	21-45	21-45	21-45#	21-45	21-45#	21-45#	21-49	21-49	21-49	21-49#
	21-49#	21-49#	21-53	21-53	21-53#	21-53#	21-53#	21-57	21-57#	21-57#	21-57	21-57#	21-57#	21-57#
	21-64	21-64	21-64#	21-64#	21-64#	21-64#	21-66	21-66	21-66#	21-66#	21-66#	21-66#	21-74	21-74
	21-74	21-74#	21-74#	21-74#	21-109	21-109	21-109	21-109#	21-109#	21-109#	60-37	60-37	60-37	60-37#
	60-37#	60-37#	60-48	60-48	60-48#	60-48#	60-48#	60-48#	60-60	60-60	60-60	60-60#	60-60#	60-60#
	60-64	60-64	60-64#	60-64#	60-64#	60-64#	60-66	60-66	60-66#	60-66#	60-66#	60-66#	60-68	60-68
	60-68	60-68#	60-68#	60-68#	60-70	60-70	60-70#	60-70#	60-70#	60-70#	60-75	60-75	60-75	60-75#
	60-75#	60-75#	60-85	60-85	60-85#	60-85#	60-85#	60-85#	60-90	60-90	60-90	60-90#	60-90#	60-90#
	68-55	68-55	68-55#	68-55#	68-55#	68-55#	68-57	68-57	68-57#	68-57#	68-57#	68-57#	68-59	68-59
	68-59	68-59#	68-59#	68-59#	68-61	68-61	68-61#	68-61#	68-61#	68-61#				
T\$ERRN	7-143#	23-38	23-38#	23-41	23-41#	23-44	23-44#	23-47	23-47#	23-50	23-50#	23-53	23-53#	23-56
	23-56#	23-59	23-59#	23-62	23-62#	23-65	23-65#	23-68	23-68#	23-71	23-71#	23-74	23-74#	23-77
	23-77#	23-80	23-80#	23-83	23-83#	25-17	25-17#	25-33	25-33#					
T\$EXCP	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#	21-57	21-57#
	21-64	21-64#	21-66	21-66#	21-74	21-74#	60-37	60-37#	60-64	60-64#	60-66	60-66#	60-68	60-68#
	60-70	60-70#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#				
T\$FLAG	58-61	58-61#	58-61#	60-16	60-16#	60-16#	60-16#	60-97	60-97#	60-97#	60-97#	60-202	60-202	60-202#
	60-202#	62-17	62-17#	62-17#	62-17#	62-47	62-47#	62-47#	62-47#	63-19	63-19#	63-19#	64-20	64-20#
	64-20#	65-78	65-78	65-78#	65-78#	65-83	65-83	65-83#	65-83#	65-135	65-135	65-135#	65-135#	
T\$FREE	69-85	70-22#												
T\$GMAN	7-143#	21-15#	21-15#	21-17#	21-17#	21-19#	21-19#	21-45#	21-45#	21-49#	21-49#	21-53#	21-53#	21-57#
	21-57#	21-64#	21-64#	21-66#	21-66#	21-74#	21-74#	60-37#	60-37#	60-64#	60-64#	60-66#	60-66#	60-68#
	60-68#	60-70#	60-70#											
T\$HILI	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#	21-57	21-57#
	21-64	21-64#	21-66	21-66#	21-74	21-74#	60-37	60-37#	60-64	60-64#	60-66	60-66#	60-68	60-68#
	60-70	60-70#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#				
T\$LAST	7-143#	69-85#	70-15											
T\$LOLI	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#	21-57	21-57#
	21-64	21-64#	21-66	21-66#	21-74	21-74#	60-37	60-37#	60-64	60-64#	60-66	60-66#	60-68	60-68#
	60-70	60-70#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#				
T\$LSYM	7-143	7-143#	9-25	10-20	15-14	15-25	15-34	15-40	15-49	58-76	60-217	61-18	62-62	63-34
	64-35	66-152	68-62	69-22										
T\$LTNO	69-85#													
T\$NEST	7-143#	7-169	7-169	7-169#	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#	57-1	57-1	57-1#	57-1#	58-41	58-41	58-41#	58-47	58-47	58-47#
	58-76	58-76	58-76	58-76#	59-8	59-8	59-8#	59-12	59-12	59-12	59-12#	60-8	60-8	60-8#
	60-217	60-217	60-217	60-217#	61-10	61-10	61-10#	61-18	61-18	61-18	61-18#	62-8	62-8	62-8#
	62-62	62-62	62-62	62-62#	63-8	63-8	63-8#	63-34	63-34	63-34	63-34#	64-9	64-9	64-9#
	64-35	64-35	64-35	64-35#	64-36	64-36	64-36#	64-36	64-36#	65-50	65-50	65-50#	65-68	65-68#

	66-152	66-152	66-152	66-152#	67-1	67-1	67-1	67-1#	68-43	68-43	68-43#	68-53	68-53	68-53#
	68-62	68-62	68-62	68-62#	69-12	69-12	69-12#	69-22	69-22	69-22	69-22#	69-86	69-86	69-86
	69-86#													
T\$NSO	7-169#	10-21	11-51#	57-1	58-41#	64-36	65-50#	67-1	68-43#	69-86				
T\$NS1	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
	58-47#	58-76	59-8#	59-12	60-8#	60-217	61-10#	61-18	62-8#	62-62	63-8#	63-34	64-9#	64-35
	65-68#	66-152	68-53#	68-62	69-12#	69-22								
T\$PCNT	70-15#	70-16	70-16	70-16#										
T\$PTAB	70-16	70-16#												
T\$PTHV	7-197	70-22#												
T\$PTNU	7-143#	70-16	70-16#	70-22	70-22									
T\$SAVL	7-143#													
T\$SEGL	7-143#													
T\$SIZE	69-85	70-22#												
T\$SUBN	7-143#	65-68#												
T\$TAGL	7-143#													
T\$TAGN	7-143#	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	15-36#	15-42	15-42	15-42#	58-47	58-47	58-47#	59-8	59-8	59-8#
	60-8	60-8	60-8#	61-10	61-10	61-10#	62-8	62-8	62-8#	63-8	63-8	63-8#	64-9	64-9
	64-9#	65-68	65-68	65-68#	68-53	68-53	68-53#	69-12	69-12	69-12#	70-15	70-15	70-15#	70-16
	70-16	70-16	70-16	70-16#	70-16#	70-16#								
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#	21-15	21-15	21-15	21-15#	21-15#	21-15#	21-17	21-17
	21-17	21-17#	21-17#	21-17#	21-19	21-19	21-19	21-19#	21-19#	21-19#	21-45	21-45	21-45	21-45#
	21-45#	21-45#	21-49	21-49	21-49	21-49#	21-49#	21-49#	21-53	21-53	21-53	21-53#	21-53#	21-53#
	21-57	21-57	21-57	21-57#	21-57#	21-57#	21-64	21-64	21-64	21-64#	21-64#	21-64#	21-66	21-66
	21-66	21-66#	21-66#	21-66#	21-74	21-74	21-74	21-74#	21-74#	21-74#	21-109	21-109	21-109	21-109#
	21-109#	21-109#	57-1	57-1#	58-61	58-61#	58-76	58-76#	59-12	59-12#	60-16	60-16#	60-37	60-37
	60-37	60-37#	60-37#	60-37#	60-48	60-48	60-48	60-48#	60-48#	60-48#	60-60	60-60	60-60	60-60#
	60-60#	60-60#	60-64	60-64	60-64	60-64#	60-64#	60-64#	60-64#	60-66	60-66	60-66	60-66#	60-66#
	60-68	60-68	60-68	60-68#	60-68#	60-68#	60-70	60-70	60-70	60-70#	60-70#	60-70#	60-75	60-75
	60-75	60-75#	60-75#	60-75#	60-85	60-85	60-85	60-85#	60-85#	60-85#	60-90	60-90	60-90	60-90#
	60-90#	60-90#	60-97	60-97#	60-202	60-202#	60-217	60-217#	61-18	61-18#	62-17	62-17#	62-47	62-47#
	62-62	62-62#	63-19	63-19#	63-34	63-34#	64-20	64-20#	64-35	64-35#	64-36	64-36#	65-78	65-78#
	65-83	65-83#	65-135	65-135#	66-152	66-152#	67-1	67-1#	68-55	68-55	68-55	68-55#	68-55#	68-55#
	68-57	68-57	68-57	68-57#	68-57#	68-57#	68-59	68-59	68-59	68-59#	68-59#	68-59#	68-61	68-61
	68-61	68-61#	68-61#	68-61#	68-62	68-62#	69-22	69-22#	69-86	69-86#	69-86#			
T\$TEST	7-143#	65-68	65-68	65-68#	69-85									
T\$TSTM	7-143#	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	18-37	18-38	18-40	18-41	18-43	18-44	18-49	18-50	19-28	21-15
	21-17	21-19	21-43	21-45	21-49	21-53	21-57	21-64	21-66	21-74	21-109	23-38	23-41	23-44
	23-47	23-50	23-53	23-56	23-59	23-62	23-65	23-68	23-71	23-74	23-77	23-80	23-83	25-12
	25-17	25-24	25-27	25-33	26-21	29-46	30-20	30-22	30-33	30-39	30-42	30-46	38-40	41-38
	45-43	45-48	45-51	45-55	46-23	46-26	46-42	46-46	46-49	46-54	58-76	60-10	60-12	60-16
	60-18	60-22	60-27	60-35	60-37	60-48	60-51	60-60	60-64	60-66	60-68	60-70	60-75	60-85
	60-90	60-96	60-97	60-99	60-202	60-217	61-18	62-10	62-14	62-17	62-21	62-24	62-27	62-32
	62-34	62-36	62-47	62-62	63-34	64-35	65-78	65-83	65-92	65-102	65-135	66-15	66-28	66-152
T\$TSTS	7-143#	65-68#												
T1	8-8	65-68#												
TABELD	26-17#	60-93												
TAP	12-130#													
TBLPTR	13-60#	27-15	27-17	27-19*	27-20	27-22*	66-58	66-120*						
TD	12-229#	13-80	13-104	13-105	28-17	33-17	33-22	33-26	48-44	53-19				
TDBCNT	13-79#	43-37	44-23	44-31										
TDCASE	32-48	33-16#												
TDOPY1	15-19	32-22	32-27	32-47	44-42*	44-43*	44-44*	44-47	69-48#					

WCKHDR	14-93#	20-20		
WLE	12-89#			
WLOCK	14-98#	25-63		
WOR	12-187#			
WRDAT	14-91#	39-20	49-21	
WRITCK	17-84	20-15#		
WRITTD	21-91	42-18	43-22#	66-95
WRL	12-69#	25-61		
WRTBSF	39-15#	65-134	66-131	
WRTDAT	12-219#	39-19	49-20	
WRTFE2	13-18#	60-109*	60-143*	65-80
WRTTD	12-221#	43-33	60-162	65-100
WRU	12-186#			
XSALWA	7-143#			
XSALS	7-143#			
XSOFFS	7-143#			
X\$TRUE	7-143#			

BCOMPL	1-15#	7-143#	60-20	60-29				
BERROR	1-19#	7-143#						
BGNAU	1-23#	7-143#	64-9					
BGNAUT	1-31#	7-143#	61-10					
BGNCLN	1-39#	7-143#	62-8					
BGNDU	1-47#	7-143#	63-8					
BGNHRD	1-55#	7-143#	68-53					
BGNHW	1-66#	7-143#	9-10					
BGNINI	1-77#	7-143#	60-8					
BGNMOD	1-85#	7-143#	7-169	11-51	58-41	65-50	68-43	
BGNMSG	1-98#	7-143#	15-9	15-16	15-27	15-36	15-42	
BGNPRO	1-106#	7-143#	59-8					
BGNPTA	1-114#	7-143#	70-16					
BGNRPT	1-144#	7-143#	58-47					
BGNSEG	1-152#	7-143#						
BGNSET	1-161#	7-143#	70-15					
BGNSFT	1-182#	7-143#	69-12					
BGNSRV	1-193#	7-143#						
BGNSUB	1-201#	7-143#						
BGNSW	1-225#	7-143#	10-10					
BGNTST	1-236#	7-143#	65-68					
BNCOMP	1-266#	7-143#	26-22	60-14	60-24	65-103		
BNERRO	1-270#	7-143#						
BREAK	1-274#	7-143#	65-92	66-28				
BRESET	1-278#	7-143#	60-10					
CKLOOP	1-282#	7-143#						
CLOCK	1-286#	7-143#						
CLOSE	1-292#	7-143#						
CLRVEC	1-296#	7-143#						
COMMEN	1-301#	7-143#						
DELAY	1-322#	7-143#	22-104					
DESCRI	1-317#	7-143#	14-27					
DEVTYP	1-341#	7-143#	14-17					
DISPAT	1-346#	7-143#	8-8					
DISPLA	1-360#	7-143#						
DOCLN	1-376#	7-143#	60-51	60-96				
DODU	1-380#	7-143#						
DORPT	1-385#	7-143#						
ENDAU	1-389#	7-143#	64-35					
ENDAUT	1-401#	7-143#	61-18					
ENDCLN	1-413#	7-143#	62-62					
ENDCOM	1-425#	7-143#						
ENDDU	1-441#	7-143#	63-34					
ENDHRD	1-453#	7-143#	68-62					
ENDHW	1-465#	7-143#	9-25					
ENDINI	1-475#	7-143#	60-217					
ENDMOD	1-487#	7-143#	10-21	57-1	64-36	67-1	69-86	
ENDMSG	1-500#	7-143#	15-14	15-25	15-34	15-40	15-49	
ENDPRO	1-512#	7-143#	59-12					
ENDPTA	1-520#	7-143#	70-21					
ENDRPT	1-529#	7-143#	58-76					
ENDSEG	1-541#	7-143#						
ENDSET	1-555#	7-143#	70-22					
ENDSFT	1-568#	7-143#	69-22					
ENDSRV	1-580#	7-143#						
ENDSUB	1-596#	7-143#						

ENDSW	1-614#	7-143#	10-20											
ENDTST	1-624#	7-143#	66-152											
EQUALS	1-642#	7-143#	11-57											
ERRDF	1-714#	7-143#	25-17	25-33										
ERRHRD	1-718#	7-143#	23-38	23-41	23-44	23-47	23-50	23-53	23-56	23-59	23-62	23-65	23-68	23-71
	23-74	23-77	23-80	23-83										
ERROR	1-722#	7-143#												
ERRSF	1-726#	7-143#												
ERRSOF	1-730#	7-143#												
ERRIBL	1-734#	7-143#												
ESCAPE	1-744#	7-143#												
EXIT	1-771#	7-143#	58-61	60-16	60-97	60-202	62-17	62-47	63-19	64-20	65-78	65-83	65-135	
FEQUAL	1-810#	7-143#												
GETBYT	1-824#	7-143#												
GETPRI	1-834#	7-143#												
GETWOR	1-829#	7-143#												
GMANIA	1-839#	7-143#												
GMANID	1-848#	7-143#	21-15	21-17	21-19	21-45	21-49	21-53	21-57	21-64	21-66	21-74	60-37	60-64
	60-66	60-68	60-70											
GMANIL	1-859#	7-143#	21-109	60-48	60-60	60-75	60-85	60-90						
GPHARD	1-868#	7-143#	26-21											
GPRMA	1-874#	7-143#	68-55	68-57										
GPRMD	1-903#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	60-37	60-37#	60-64	60-64#	60-66	60-66#
	60-68	60-68#	60-70	60-70#	68-59	68-61								
GPRML	1-934#	7-143#	21-109	21-109#	60-48	60-48#	60-60	60-60#	60-75	60-75#	60-85	60-85#	60-90	60-90#
HEADER	1-954#	7-143#	7-197											
INLOOP	1-962#	7-143#												
IOSETU	1-966#	7-143#												
IOSTAR	1-974#	7-143#												
KT11	1-982#	7-143#												
LASTAD	1-:47#	7-143#	69-85											
M\$BYTE	1-D00#	7-143#	7-197	7-197	7-197	7-197#								
M\$CHEC	1-E18#	7-143#	58-61	58-61#	60-16	60-16#	60-97	60-97#	60-202	60-202#	62-17	62-17#	62-47	62-47#
	63-19	63-19#	64-20	64-20#	65-78	65-78#	65-83	65-83#	65-135	65-135#				
M\$CNTO	1-E82#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	60-37	60-37#	60-48	60-48#
	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85	60-85#
	60-90	60-90#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#				
M\$COUN	1-D66#	7-143#	15-13	15-13#	15-19	15-19	15-19	15-19#	15-21	15-21	15-21	15-21#	15-24	15-24#
	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#
	15-45	15-45	15-45#	15-48	15-48#	18-16	18-16#	18-20	18-20	18-20#	18-37	18-37#	18-38	18-38
	18-38	18-38	18-38	18-38	18-38	18-38#	18-40	18-40#	18-41	18-41	18-41	18-41#	18-41	18-41
	18-41	18-41#	18-43	18-43#	18-44	18-44	18-44	18-44#	18-44	18-44	18-44	18-44#	18-49	18-49#
	18-50	18-50	18-50#	19-28	19-28#	21-43	21-43#	25-12	25-12#	25-24	25-24#	25-27	25-27#	29-46
	29-46#	30-20	30-20#	30-22	30-22#	30-33	30-33	30-33	30-33	30-33	30-33	30-33#	30-39	30-39#
	30-42#	30-46	30-46#	38-40	38-40#	41-38	41-38	41-38	41-38#	45-43	45-43	45-43#	45-48	45-48#
	45-51	45-51#	45-55	45-55#	46-23	46-23#	46-26	46-26#	46-42	46-42	46-42	46-42#	46-42	46-42#
	46-46	46-46#	46-49	46-49#	46-54	46-54#	60-35	60-35#	62-14	62-14#	62-21	62-21	62-21#	62-24
	62-24	62-24#	62-27	62-27	62-27#	62-32	62-32	62-32#	62-34	62-34#	62-36	62-36	62-36#	66-15
	66-15#													
M\$DATA	1-B67#	7-143#	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197#	7-197#	14-17
	14-17#	14-27	14-27#											
M\$DECR	1-D29#	7-143#	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#	57-1	57-1#	58-76	58-76#	59-12	59-12#	60-217	60-217#	61-18	61-18#
	62-62	62-62#	63-34	63-34#	64-35	64-35#	64-36	64-36#	66-152	66-152#	67-1	67-1#	68-62	68-62#
	69-22	69-22#	69-86	69-86#	70-16	70-16#								
M\$DEFA	1-E70#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	60-37	60-37#	60-48	60-48#
	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85	60-85#
	60-90	60-90#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#				
M\$ENDE	1-D74#	7-143#	9-25#	10-20#	10-21#	15-14#	15-25#	15-34#	15-40#	15-49#	57-1#	58-76#	60-217#	61-18#
	62-62#	63-34#	64-35#	64-36#	66-152#	67-1#	68-62#	69-22#	69-86#					
M\$ERRI	1-D49#	7-143#	23-38	23-38#	23-41	23-41#	23-44	23-44#	23-47	23-47#	23-50	23-50#	23-53	23-53#
	23-56	23-56#	23-59	23-59#	23-62	23-62#	23-65	23-65#	23-68	23-68#	23-71	23-71#	23-74	23-74#
	23-77	23-77#	23-80	23-80#	23-83	23-83#	25-17	25-17#	25-33	25-33#				
M\$ESCA	1-D06#	7-143#												
M\$ESCS	1-D10#	7-143#												
M\$EXCP	1-E01#	7-143#	21-15	21-15#	21-15#	21-17	21-17	21-17#	21-19	21-19	21-19#	21-45	21-45	21-45#
	21-49	21-49#	21-49#	21-53	21-53	21-53#	21-57	21-57#	21-57#	21-64	21-64#	21-64#	21-66	21-66#
	21-66#	21-74	21-74	21-74#	60-37	60-37#	60-37#	60-64	60-64#	60-64#	60-66	60-66#	60-66#	60-68
	60-68	60-68#	60-70	60-70#	60-70#	68-55	68-55	68-55#	68-57	68-57#	68-57#	68-59	68-59#	68-59#
	68-61	68-61#	68-61#											
M\$EXIT	1-D14#	7-143#	58-61#	60-16	60-16#	60-97	60-97#	60-202	60-202#	62-17	62-17#	62-47	62-47#	63-19#
	64-20#	65-78	65-78#	65-83	65-83#	65-135	65-135#							
M\$EXSE	1-D22#	7-143#	58-61#	60-16#	60-97#	60-202#	62-17#	62-47#	63-19#	64-20#	65-78#	65-83#	65-135#	
M\$EXTJ	1-D18#	7-143#	58-61	58-61#	60-16#	60-97#	60-202#	62-17#	62-47#	63-19	63-19#	64-20	64-20#	65-78#
	65-83#	65-135#												
M\$GEN	1-D38#	7-143#	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	9-25	9-25#	10-10	10-10	10-10#	10-20	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#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#
	21-53	21-53#	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	58-47	58-47#
	58-76	58-76#	59-8	59-8#	60-8	60-8#	60-37	60-37#	60-48	60-48#	60-60	60-60#	60-64	60-64#
	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85	60-85#	60-90	60-90#	60-217	60-217#
	61-10	61-10#	61-18	61-18#	62-8	62-8#	62-62	62-62#	63-8	63-8#	63-34	63-34#	64-9	64-9#
	64-35	64-35#	65-68	65-68#	66-152	66-152#	68-53	68-53#	68-62	68-62#	69-12	69-12#	69-22	69-22#
	69-85	69-85#	70-16	70-16#	70-21	70-21#								
M\$GENE	1-C38#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53	21-53#
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	60-37	60-37#	60-48	60-48#
	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85	60-85#
	60-90	60-90#												
M\$GETS	1-D35#	7-143#	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#	57-1	57-1#	58-76	58-76#	59-12	59-12#	60-217	60-217#	61-18	61-18#
	62-62	62-62#	63-34	63-34#	64-35	64-35#	64-36	64-36#	66-152	66-152#	67-1	67-1#	68-62	68-62#
	69-22	69-22#	69-86	69-86#										
M\$GETT	1-B77#	7-143#	58-61#	60-16#	60-97#	60-202#	62-17#	62-47#	63-19#	64-20#	65-78#	65-83#	65-135#	
M\$GNGB	1-C02#	7-143#	7-169#	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197	7-197
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#	7-197#
	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#	58-41#	58-47	58-47#	59-8	59-8#	60-8	60-8#	61-10	61-10#	62-8

	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33
	30-33#	30-33#	30-33#	30-33#	30-39	30-39	30-39	30-39	30-39	30-39	30-39#	30-39#	30-39#
	30-39#	30-42	30-42	30-42	30-42	30-42	30-42#	30-42#	30-42#	30-42#	30-42#	30-42#	30-46
	30-46	30-46	30-46	30-46#	30-46#	30-46#	38-40	38-40	38-40	38-40	38-40	38-40	38-40#
	38-40#	38-40#	38-40#	38-40#	41-38	41-38	41-38	41-38	41-38	41-38	41-38	41-38	41-38#
	41-38#	41-38#	41-38#	41-38#	41-38#	45-43	45-43	45-43	45-43	45-43	45-43	45-43	45-43#
	45-43#	45-43#	45-43#	45-43#	45-48	45-48	45-48	45-48	45-48	45-48	45-48#	45-48#	45-48#
	45-48#	45-51	45-51	45-51	45-51	45-51	45-51#	45-51#	45-51#	45-51#	45-51#	45-51#	45-55
	45-55	45-55	45-55	45-55#	45-55#	45-55#	46-23	46-23	46-23	46-23	46-23	46-23	46-23#
	46-23#	46-23#	46-23#	46-23#	46-26	46-26	46-26	46-26	46-26	46-26#	46-26#	46-26#	46-42
	46-42	46-42	46-42	46-42	46-42	46-42	46-42	46-42	46-42	46-42#	46-42#	46-42#	46-42#
	46-42#	46-42#	46-42#	46-42#	46-46	46-46	46-46	46-46	46-46	46-46#	46-46#	46-46#	46-46#
	46-46#	46-49	46-49	46-49	46-49	46-49	46-49#	46-49#	46-49#	46-49#	46-49#	46-49#	46-54
	46-54	46-54	46-54	46-54#	46-54#	46-54#	58-61	58-61	58-61#	58-61#	58-61#	58-61#	60-10
	60-10#	60-12	60-12	60-12#	60-12#	60-14	60-16	60-16	60-16#	60-16#	60-16#	60-16#	60-18#
	60-18#	60-20	60-20#	60-22	60-22	60-22#	60-24	60-24#	60-27	60-27	60-27#	60-27#	60-29
	60-29#	60-35	60-35	60-35	60-35	60-35#	60-35#	60-35#	60-35#	60-35#	60-37	60-37	60-37
	60-37	60-37	60-37	60-37	60-37#	60-37#	60-37#	60-37#	60-37#	60-37#	60-37	60-37	60-37
	60-48#	60-48#	60-48#	60-48#	60-51	60-51#	60-60	60-60	60-60	60-60	60-60	60-60	60-60#
	60-60#	60-60#	60-64	60-64	60-64	60-64	60-64	60-64	60-64	60-64	60-64#	60-64#	60-64#
	60-66	60-66	60-66	60-66	60-66	60-66	60-66	60-66#	60-66#	60-66#	60-66#	60-66#	60-68
	60-68	60-68	60-68	60-68	60-68	60-68	60-68#	60-68#	60-68#	60-68#	60-68#	60-68#	60-68
	60-70	60-70	60-70	60-70	60-70#	60-70#	60-70#	60-70#	60-70#	60-70#	60-70	60-70	60-70
	60-75#	60-75#	60-75#	60-75#	60-85	60-85	60-85	60-85	60-85	60-85	60-85#	60-85#	60-85#
	60-90	60-90	60-90	60-90	60-90	60-90	60-90#	60-90#	60-90#	60-90#	60-96	60-96#	60-97
	60-97#	60-97#	60-99	60-99	60-99#	60-99#	60-202	60-202	60-202#	60-202#	60-217	60-217#	61-18
	62-10	62-10	62-10#	62-10#	62-14	62-14	62-14	62-14	62-14#	62-14#	62-14#	62-14#	62-17
	62-17	62-17#	62-17#	62-21	62-21	62-21	62-21	62-21	62-21	62-21#	62-21#	62-21#	62-21#
	62-21#	62-21#	62-24	62-24	62-24	62-24	62-24	62-24	62-24#	62-24#	62-24#	62-24#	62-24#
	62-24#	62-27	62-27	62-27	62-27	62-27	62-27	62-27	62-27#	62-27#	62-27#	62-27#	62-27#
	62-32	62-32	62-32	62-32	62-32	62-32	62-32#	62-32#	62-32#	62-32#	62-32#	62-32#	62-34
	62-34	62-34	62-34	62-34	62-34	62-34#	62-34#	62-34#	62-34#	62-34#	62-36	62-36	62-36
	62-36	62-36	62-36	62-36#	62-36#	62-36#	62-36#	62-36#	62-36#	62-36#	62-36	62-36	62-36
	62-62#	63-19	63-19	63-19#	63-19#	63-34	63-34#	63-34#	64-20	64-20	64-20#	64-20#	64-35
	65-78	65-78#	65-78#	65-83	65-83	65-83#	65-83#	65-83#	65-92	65-92#	65-102	65-102#	65-103
	65-135	65-135#	65-135#	66-15	66-15	66-15	66-15	66-15	66-15	66-15#	66-15#	66-15#	66-15#
	66-28	66-28#	66-152	66-152#	68-53	68-53#	68-55	68-55	68-55	68-55	68-55#	68-57	68-57
	68-57	68-57#	68-59	68-59	68-59	68-59	68-59#	68-59#	68-61	68-61	68-61	68-61	68-61#
	68-62	68-62#	69-12	69-12#	69-22	69-22#	69-85	69-85	69-85	69-85#	70-16	70-16	70-16#
MSGNLS	1-C13#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	60-37	60-37#	60-48
	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85
	60-90	60-90#											
MSGNSU	1-B98#	7-143#											
MSGNTA	1-B90#	7-143#	9-25	9-25#	10-20	10-20#	15-14	15-14#	15-25	15-25#	15-34	15-34#	15-40
	15-49	15-49#	58-76	58-76#	60-217	60-217#	61-18	61-18#	62-62	62-62#	63-34	63-34#	64-35
	66-152	66-152#	68-62	68-62#	69-22	69-22#	70-16	70-16#	70-21	70-21#			
MSGNTE	1-B94#	7-143#	65-68	65-68#									
M\$HAPT	1-A39#	7-143#	7-197	7-197#									
M\$HNAP	1-B24#	7-143#	7-197	7-197#									
M\$INCR	1-D26#	7-143#	7-169	7-169#	9-10	9-10	9-10#	9-10#	10-10	10-10	10-10#	10-10#	11-51
	15-9	15-9	15-9#	15-9#	15-13#	15-14#	15-16	15-16	15-16#	15-16#	15-19#	15-21#	15-25#
	15-27	15-27	15-27#	15-27#	15-30#	15-31#	15-33#	15-34#	15-36	15-36	15-36#	15-36#	15-39#
	15-42	15-42	15-42#	15-42#	15-45#	15-48#	15-49#	18-16#	18-20#	18-37#	18-38#	18-40#	18-41#
	18-44#	18-49#	18-50#	19-28#	21-15	21-15#	21-15#	21-17	21-17#	21-17#	21-19	21-19#	21-19#
	21-45	21-45#	21-45#	21-49	21-49#	21-49#	21-53	21-53#	21-53#	21-57	21-57#	21-57#	21-64

	21-64#	21-66	21-66#	21-66#	21-74	21-74#	21-74#	21-109	21-109#	21-109#	23-38#	23-41#	23-44#	23-47#
	23-50#	23-53#	23-56#	23-59#	23-62#	23-65#	23-68#	23-71#	23-74#	23-77#	23-80#	23-83#	25-12#	25-17#
	25-24#	25-27#	25-33#	26-21#	29-46#	30-20#	30-22#	30-33#	30-39#	30-42#	30-46#	38-40#	41-38#	45-43#
	45-48#	45-51#	45-55#	46-23#	46-26#	46-42#	46-46#	46-49#	46-54#	58-41	58-41#	58-47	58-47	58-47#
	58-47#	58-76#	59-8	59-8	59-8#	59-8#	60-8	60-8	60-8#	60-8#	60-10#	60-12#	60-16#	60-18#
	60-22#	60-27#	60-35#	60-37	60-37#	60-37#	60-48	60-48#	60-48#	60-51#	60-60	60-60#	60-60#	60-64
	60-64#	60-64#	60-66	60-66#	60-66#	60-68	60-68#	60-68#	60-70	60-70#	60-70#	60-75	60-75#	60-75#
	60-85	60-85#	60-85#	60-90	60-90#	60-90#	60-96#	60-97#	60-99#	60-202#	60-217#	61-10	61-10	61-10#
	61-10#	61-18#	62-8	62-8	62-8#	62-8#	62-10#	62-14#	62-17#	62-21#	62-24#	62-27#	62-32#	62-34#
	62-36#	62-47#	62-62#	63-8	63-8	63-8#	63-8#	63-34#	64-9	64-9	64-9#	64-9#	64-35#	65-50
	65-50#	65-68	65-68	65-68	65-68#	65-68#	65-68#	65-78#	65-83#	65-92#	65-102#	65-135#	66-15#	66-28#
	66-152#	68-43	68-43#	68-53	68-53	68-53#	68-53#	69-12	69-12	69-12#	69-12#	70-15	70-15#	70-16
	70-16	70-16	70-16#											
MSIOSE	1-A00#	7-143#												
MSLDRO	1-C42#	7-143#	26-21	26-21#	60-12	60-12#	60-18	60-18#	60-22	60-22#	60-27	60-27#	60-99	60-99#
	62-10	62-10#												
MSMASK	1-a71#	7-143#												
MSMCHI	1-4#	7-143	7-143#	7-143#										
MSMCLO	1-a24#	7-143	7-143#	7-143#										
MSMSK1	1-a77#	7-143#												
MSPOP	1-B81#	7-143#	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#	57-1	57-1#	58-76	58-76#	59-12	59-12#	60-217	60-217#	61-18	61-18#
	62-62	62-62#	63-34	63-34#	64-35	64-35#	64-36	64-36#	66-152	66-152#	67-1	67-1#	68-62	68-62#
	69-22	69-22#	69-86	69-86#										
MSPRIN	1-a36#	7-143#	15-13	15-13#	15-19	15-19#	15-21	15-21#	15-24	15-24#	15-30	15-30#	15-31	15-31#
	15-33	15-33#	15-39	15-39#	15-45	15-45#	15-48	15-48#	18-16	18-16#	18-20	18-20#	18-37	18-37#
	18-38	18-38#	18-40	18-40#	18-41	18-41#	18-43	18-43#	18-44	18-44#	18-49	18-49#	18-50	18-50#
	19-28	19-28#	21-43	21-43#	25-12	25-12#	25-24	25-24#	25-27	25-27#	29-46	29-46#	30-20	30-20#
	30-22	30-22#	30-33	30-33#	30-39	30-39#	30-42	30-42#	30-46	30-46#	38-40	38-40#	41-38	41-38#
	45-43	45-43#	45-48	45-48#	45-51	45-51#	45-55	45-55#	46-23	46-23#	46-26	46-26#	46-42	46-42#
	46-46	46-46#	46-49	46-49#	46-54	46-54#	60-35	60-35#	62-14	62-14#	62-21	62-21#	62-24	62-24#
	62-27	62-27#	62-32	62-32#	62-34	62-34#	62-36	62-36#	66-15	66-15#				
MSPUSH	1-a31#	7-143#	7-169	7-169#	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#	58-41	58-41#	58-47	58-47#	59-8	59-8#	60-8	60-8#
	61-10	61-10#	62-8	62-8#	63-8	63-8#	64-9	64-9#	65-50	65-50#	65-68	65-68#	68-43	68-43#
	68-53	68-53#	69-12	69-12#										
MSPUT	1-C72#	7-143#	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#	18-20	18-20	18-20#	18-20	18-20	18-20#
	18-37	18-37#	18-37#	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38	18-38#	18-40
	18-40	18-40#	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-43	18-43
	18-43#	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44	18-44#	18-49	18-49	18-49#
	18-50	18-50	18-50	18-50	18-50#	19-28	19-28	19-28	19-28#	21-43	21-43	21-43#	25-12	25-12
	25-12	25-12#	25-24	25-24	25-24#	25-27	25-27	25-27#	29-46	29-46	29-46	29-46#	30-20	30-20
	30-20	30-20#	30-22	30-22	30-22#	30-33	30-33	30-33	30-33	30-33	30-33	30-33	30-33#	30-39
	30-39	30-39	30-39#	30-42	30-42	30-42	30-42#	30-46	30-46	30-46#	38-40	38-40	38-40	38-40#
	41-38	41-38	41-38	41-38	41-38	41-38#	45-43	45-43	45-43	45-43	45-43#	45-48	45-48	45-48
	45-48#	45-51	45-51	45-51	45-51#	45-55	45-55	45-55#	46-23	46-23	46-23	46-23#	46-26	46-26
	46-26#	46-42	46-42	46-42	46-42	46-42	46-42	46-42	46-42#	46-46	46-46	46-46	46-46#	46-49
	46-49	46-49	46-49#	46-54	46-54	46-54#	60-35	60-35	60-35#	62-14	62-14	62-14#	62-21	62-21
	62-21	62-21	62-21#	62-24	62-24	62-24	62-24	62-24	62-24#	62-27	62-27	62-27	62-27#	62-32
	62-32	62-32	62-32	62-32#	62-34	62-34	62-34	62-34	62-34#	62-36	62-36	62-36	62-36#	66-15
	66-15	66-15	66-15#											
M\$PUT1	1-C81#	7-143#	15-13	15-13	15-13#	15-19	15-19	15-19	15-19	15-19	15-19	15-19#	15-19#	15-19#
	15-19#	15-19#	15-21	15-21	15-21	15-21	15-21	15-21	15-21#	15-21#	15-21#	15-21#	15-24	15-24

	15-24#	15-24#	15-30	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#	15-31#	15-31#	15-31#	15-31#	15-31#	15-31#	15-31#	15-33	15-33	15-33#
	15-39	15-39	15-39#	15-39#	15-45	15-45	15-45	15-45	15-45#	15-45#	15-45#	15-45#	15-48
	15-48#	15-48#	18-16	18-16	18-16	18-16#	18-16#	18-16#	18-20	18-20	18-20	18-20#	18-20#
	18-20#	18-20#	18-37	18-37	18-37#	18-37#	18-38	18-38	18-38	18-38	18-38	18-38	18-38
	18-38	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-38#	18-40	18-40	18-40#
	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41	18-41#	18-41#	18-41#	18-41#	18-41#
	18-41#	18-41#	18-41#	18-41#	18-43	18-43	18-43#	18-43#	18-44	18-44	18-44	18-44	18-44
	18-44	18-44	18-44	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-44#	18-49
	18-49#	18-49#	18-50	18-50	18-50	18-50	18-50#	18-50#	18-50#	18-50#	19-28	19-28	19-28#
	19-28#	19-28#	21-43	21-43	21-43#	21-43#	25-12	25-12	25-12	25-12#	25-12#	25-12#	25-24
	25-24#	25-24#	25-27	25-27	25-27#	25-27#	29-46	29-46	29-46	29-46#	29-46#	29-46#	30-20
	30-20	30-20#	30-20#	30-20#	30-22	30-22	30-22#	30-22#	30-33	30-33	30-33	30-33	30-33
	30-33	30-33#	30-33#	30-33#	30-33#	30-33#	30-33#	30-33#	30-39	30-39	30-39	30-39#	30-39#
	30-42	30-42	30-42	30-42#	30-42#	30-42#	30-46	30-46	30-46#	30-46#	38-40	38-40	38-40#
	38-40#	38-40#	41-38	41-38	41-38	41-38	41-38	41-38#	41-38#	41-38#	41-38#	41-38#	45-43
	45-43	45-43	45-43#	45-43#	45-43#	45-43#	45-48	45-48	45-48	45-48#	45-48#	45-48#	45-51
	45-51	45-51#	45-51#	45-51#	45-55	45-55	45-55#	45-55#	46-23	46-23	46-23	46-23#	46-23#
	46-26	46-26	46-26#	46-26#	46-42	46-42	46-42	46-42	46-42	46-42	46-42	46-42#	46-42#
	46-42#	46-42#	46-42#	46-42#	46-46	46-46	46-46	46-46#	46-46#	46-46#	46-49	46-49	46-49#
	46-49#	46-49#	46-54	46-54	46-54#	46-54#	60-35	60-35	60-35#	60-35#	62-14	62-14	62-14#
	62-21	62-21	62-21	62-21	62-21#	62-21#	62-21#	62-21#	62-24	62-24	62-24	62-24	62-24#
	62-24#	62-24#	62-27	62-27	62-27	62-27	62-27#	62-27#	62-27#	62-27#	62-32	62-32	62-32
	62-32#	62-32#	62-32#	62-32#	62-34	62-34	62-34	62-34#	62-34#	62-34#	62-36	62-36	62-36
	62-36#	62-36#	62-36#	62-36#	66-15	66-15	66-15	66-15#	66-15#	66-15#			
MSRADI	1-D77#	7-143#	21-15	21-15#	21-17	21-17#	21-19	21-19#	21-45	21-45#	21-49	21-49#	21-53
	21-57	21-57#	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	60-37	60-37#	60-48
	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85
	60-90	60-90#	68-55	68-55#	68-57	68-57#	68-59	68-59#	68-61	68-61#			
MSRBRO	1-C52#	7-143#											
MSRNRO	1-C62#	7-143#	26-21	26-21#									
MSSETS	1-D32#	7-143#	7-169	7-169#	9-10	9-10#	10-10	10-10#	11-51	11-51#	15-9	15-9#	15-16
	15-27	15-27#	15-36	15-36#	15-42	15-42#	58-41	58-41#	58-47	58-47#	59-8	59-8#	60-8
	61-10	61-10#	62-8	62-8#	63-8	63-8#	64-9	64-9#	65-50	65-50#	65-68	65-68#	68-43
	68-53	68-53#	69-12	69-12#									
MSSTAR	1-A33#	7-143#											
MSVC	1-C33#	7-143#	15-13	15-13#	15-14	15-14#	15-19	15-19#	15-21	15-21#	15-24	15-24#	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
	15-48	15-48#	15-49	15-49#	18-16	18-16#	18-20	18-20#	18-37	18-37#	18-38	18-38#	18-40
	18-41	18-41#	18-43	18-43#	18-44	18-44#	18-49	18-49#	18-50	18-50#	19-28	19-28#	21-15
	21-17	21-17#	21-19	21-19#	21-43	21-43#	21-45	21-45#	21-49	21-49#	21-53	21-53#	21-57
	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	23-38	23-41	23-44	23-47	23-50
	23-56	23-59	23-62	23-65	23-68	23-71	23-74	23-77	23-80	23-83	25-12	25-12#	25-17
	25-24#	25-27	25-27#	25-33	26-21	26-21#	29-46	29-46#	30-20	30-20#	30-22	30-22#	30-33
	30-39	30-39#	30-42	30-42#	30-46	30-46#	38-40	38-40#	41-38	41-38#	45-43	45-43#	45-48
	45-51	45-51#	45-55	45-55#	46-23	46-23#	46-26	46-26#	46-42	46-42#	46-46	46-46#	46-49
	46-54	46-54#	58-61#	58-76	58-76#	60-10	60-10#	60-12	60-12#	60-16	60-16#	60-18	60-22
	60-22#	60-27	60-27#	60-35	60-35#	60-37	60-37#	60-48	60-48#	60-51	60-51#	60-60	60-64
	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#	60-75	60-75#	60-85	60-85#	60-90	60-96
	60-96#	60-97	60-97#	60-99	60-99#	60-202	60-202#	60-217	60-217#	61-18	61-18#	62-10	62-14
	62-14#	62-17	62-17#	62-21	62-21#	62-24	62-24#	62-27	62-27#	62-32	62-32#	62-34	62-36
	62-36#	62-47	62-47#	62-62	62-62#	63-19#	63-34	63-34#	64-20#	64-35	64-35#	65-78	65-83
	65-83#	65-92	65-92#	65-102	65-102#	65-135	65-135#	66-15	66-15#	66-28	66-28#	66-152	66-152#
MSLAB	1-C29#	7-143#	15-13#	15-14#	15-19#	15-21#	15-24#	15-25#	15-30#	15-31#	15-33#	15-34#	15-39#
	15-45#	15-48#	15-49#	18-16#	18-20#	18-37#	18-38#	18-40#	18-41#	18-43#	18-44#	18-49#	18-50#
	21-15#	21-17#	21-19#	21-43#	21-45#	21-49#	21-53#	21-57#	21-64#	21-66#	21-74#	21-109#	23-38#

	23-44#	23-47#	23-50#	23-53#	23-56#	23-59#	23-62#	23-65#	23-68#	23-71#	23-74#	23-77#	23-80#	23-83#
	25-12#	25-17#	25-24#	25-27#	25-33#	26-21#	29-46#	30-20#	30-22#	30-33#	30-39#	30-42#	30-46#	38-40#
	41-38#	45-43#	45-48#	45-51#	45-55#	46-23#	46-26#	46-42#	46-46#	46-49#	46-54#	58-76#	60-10#	60-12#
	60-16#	60-18#	60-22#	60-27#	60-35#	60-37#	60-48#	60-51#	60-60#	60-64#	60-66#	60-68#	60-70#	60-75#
	60-85#	60-90#	60-96#	60-97#	60-99#	60-202#	60-217#	61-18#	62-10#	62-14#	62-17#	62-21#	62-24#	62-27#
	62-32#	62-34#	62-36#	62-47#	62-62#	63-34#	64-35#	65-78#	65-83#	65-92#	65-102#	65-135#	66-15#	66-28#
	66-152#													
M&TSTL	1-C21#	7-143#	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#	18-37	18-37#	18-38	18-38#	18-40	18-40#
	18-41	18-41#	18-43	18-43#	18-44	18-44#	18-49	18-49#	18-50	18-50#	19-28	19-28#	21-15	21-15#
	21-17	21-17#	21-19	21-19#	21-43	21-43#	21-45	21-45#	21-49	21-49#	21-53	21-53#	21-57	21-57#
	21-64	21-64#	21-66	21-66#	21-74	21-74#	21-109	21-109#	23-38	23-38#	23-38#	23-41	23-41#	23-41#
	23-44	23-44#	23-44#	23-47	23-47#	23-47#	23-50	23-50#	23-50#	23-53	23-53#	23-53#	23-56	23-56#
	23-56#	23-59	23-59#	23-59#	23-62	23-62#	23-62#	23-65	23-65#	23-65#	23-68	23-68#	23-68#	23-71
	23-71#	23-71#	23-74	23-74#	23-74#	23-77	23-77#	23-77#	23-80	23-80#	23-80#	23-83	23-83#	23-83#
	25-12	25-12#	25-17	25-17#	25-17#	25-24	25-24#	25-27	25-27#	25-33	25-33#	25-33#	26-21	26-21#
	29-46	29-46#	30-20	30-20#	30-22	30-22#	30-33	30-33#	30-39	30-39#	30-42	30-42#	30-46	30-46#
	38-40	38-40#	41-38	41-38#	45-43	45-43#	45-48	45-48#	45-51	45-51#	45-55	45-55#	46-23	46-23#
	46-26	46-26#	46-42	46-42#	46-46	46-46#	46-49	46-49#	46-54	46-54#	58-76	58-76#	60-10	60-10#
	60-12	60-12#	60-16	60-16#	60-18	60-18#	60-22	60-22#	60-27	60-27#	60-35	60-35#	60-37	60-37#
	60-48	60-48#	60-51	60-51#	60-60	60-60#	60-64	60-64#	60-66	60-66#	60-68	60-68#	60-70	60-70#
	60-75	60-75#	60-85	60-85#	60-90	60-90#	60-96	60-96#	60-97	60-97#	60-99	60-99#	60-202	60-202#
	60-217	60-217#	61-18	61-18#	62-10	62-10#	62-14	62-14#	62-17	62-17#	62-21	62-21#	62-24	62-24#
	62-27	62-27#	62-32	62-32#	62-34	62-34#	62-36	62-36#	62-47	62-47#	62-62	62-62#	63-34	63-34#
	64-35	64-35#	65-78	65-78#	65-83	65-83#	65-92	65-92#	65-102	65-102#	65-135	65-135#	66-15	66-15#
	66-28	66-28#	66-152	66-152#										
MSWORD	1-C94#	7-143#	7-197	7-197#	8-8	8-8	8-8#	21-15	21-15	21-15#	21-15#	21-17	21-17	21-17#
	21-17#	21-19	21-19	21-19#	21-19#	21-45	21-45	21-45#	21-45#	21-49	21-49	21-49#	21-49#	21-53
	21-53	21-53#	21-53#	21-57	21-57	21-57#	21-57#	21-64	21-64	21-64#	21-64#	21-66	21-66	21-66#
	21-66#	21-74	21-74	21-74#	21-74#	21-109	21-109	21-109#	21-109#	23-38	23-38	23-38	23-38#	23-41
	23-41	23-41	23-41#	23-44	23-44	23-44	23-44#	23-47	23-47	23-47	23-47#	23-50	23-50	23-50
	23-50#	23-53	23-53	23-53	23-53#	23-56	23-56	23-56#	23-56#	23-59	23-59	23-59	23-59#	23-62
	23-62	23-62	23-62#	23-65	23-65	23-65	23-65#	23-68	23-68	23-68	23-68#	23-71	23-71	23-71
	23-71#	23-74	23-74	23-74	23-74#	23-77	23-77	23-77	23-77#	23-80	23-80	23-80	23-80#	23-83
	23-83	23-83	23-83#	25-17	25-17	25-17	25-17#	25-33	25-33	25-33	25-33#	58-61	58-61#	60-16#
	60-37	60-37	60-37#	60-37#	60-48	60-48	60-48#	60-48#	60-48#	60-60	60-60	60-60#	60-60#	60-64
	60-64#	60-64#	60-66	60-66	60-66#	60-66#	60-68	60-68	60-68#	60-68#	60-70	60-70	60-70#	60-70#
	60-75	60-75	60-75#	60-75#	60-85	60-85	60-85#	60-85#	60-85#	60-90	60-90	60-90#	60-90#	60-202#
	62-17#	62-47#	63-19	63-19#	64-20	64-20#	65-78#	65-83#	65-135#	68-55	68-55#	68-57	68-57#	68-59
	68-59#	68-61	68-61#	70-16	70-16#									
MSXFER	1-@82#	7-143#												
MANUAL	1-:62#	7-143#	65-102											
MEMORY	1-:66#	7-143#												
OPEN	1-:71#	7-143#												
POINTE	1-:76#	7-143#	7-175											
POP	7-111#	20-56	21-107	22-90	23-32	24-23	26-48	28-26	28-52	29-67	30-47	34-67	36-25	38-56
	39-50	41-40	43-45	44-33	44-62	45-58	52-24	53-27	53-49	53-64	53-82	53-85	53-103	54-24
	54-74	54-109	54-125	54-258	55-25	55-48	55-71	55-96	60-176	66-114				
PRINTB	1-<39#	7-143#	15-13	15-19	15-21	15-24	15-30	15-31	15-33	15-39	15-45	15-48	18-16	18-20
	21-43													
PRINTF	1-<79#	7-143#	19-28	25-12	25-24	25-27	29-46	30-20	30-22	30-33	30-39	30-42	30-46	38-40
	41-38	45-43	45-48	45-51	45-55	46-23	46-26	46-42	46-46	46-49	46-54	60-35	62-14	62-21
	62-24	62-27	62-32	62-34	62-36	66-15								
PRINTS	1-=19#	7-143#												
PRINTX	1-=59#	7-143#	18-37	18-38	18-40	18-41	18-43	18-44	18-49	18-50				

