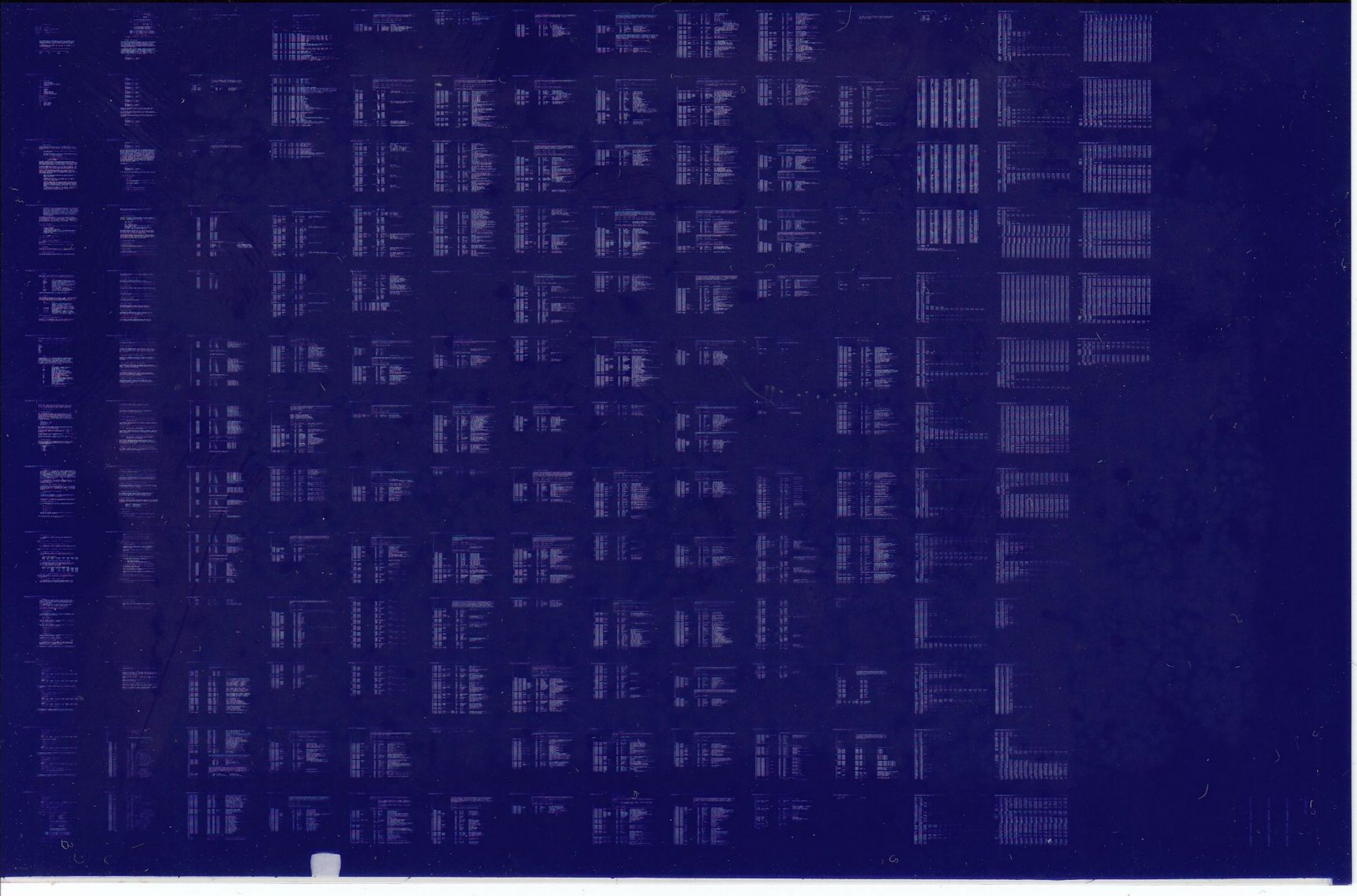
RP07

RP07 FMTR/SCANNER
CZRJKBO

AH-F957B-MC
FICHE 1 OF 1

APR 1984
digital
Made In USA



```
.REM @
```

IDENTIFICATION

PRODUCT CODE: AC-F9568-MC

PRODUCT NAME: CZRJKB0 RP07 FORMAT/SCANNER

PRODUCT DATE: DECEMBER 1, 1983

MAINTAINER: CX DIAGNOSTIC ENGINEERING

AUTHOR: MIKE LEAVITT


```
@
```

.REM @

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 PROGRAM ABSTRACT

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

A) THE FORMAT PROCESS USES MEDIA "DEFECT SKIPS" IN ADDITION TO BAD SECTOR FLAGGING.

B) THE FORMATTER PROGRAM, A STAND ALONE PROGRAM, CAN BE USED IN THE FIELD, BUT NOT INDISCRIMINATELY. THIS PROGRAM IS INTENDED FOR USE IN 2 DIFFERENT ENVIRONMENTS:

1. AT THE VENDOR'S MANUFACTURING SITE

2. AT THE CUSTOMERS SITE, USED BY TRAINED DEC FIELD SERVICE PERSONNEL

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

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

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

A) GIVEN THAT THE TRACK DESCRIPTORS ARE ALREADY WRITTEN ON THE DRIVE, THE PROGRAM WILL MAP THE STARTING LOCATIONS OF ALL NEEDED DEFECT SKIPS.

B) DEPENDENT UPON USER INPUT, THE PROGRAM WILL THEN FORMAT OR FORMAT VERIFY EITHER A SELECTED TRACK, PART OF A DRIVE'S SURFACE OR AN ENTIRE DRIVE.

C) ASSUMING A HIGH DEGREE OF SOPHISTICATION ON THE PART OF THE USER, THE PROGRAM WILL ALLOW MANUAL ENTRY OF TRACK DESCRIPTOR INFORMATION, THEN REWRITE A GIVEN TRACK DESCRIPTOR, PER USER INPUT VIA THE CONSOLE KEYBOARD. IMPLIED HERE IS THE NEED TO REFORMAT THE SPECIFIED TRACK BASED ON THE NEW TRACK DESCRIPTOR INPUT.
IF RUNNING THE FIELD VERSION OF THIS PROGRAM, WARNING MESSAGES WILL BE GENERATED TO INSURE THAT THE USER HAS A CHANCE TO CHANGE FUNCTIONS AND AVOID A POSSIBLE CATASTROPHY, I.E. DESTRUCTION OF A SYSTEM DATA FILE.

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

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

1.2 SYSTEM REQUIREMENTS

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

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

1.3 RELATED DOCUMENTS AND STANDARDS

XXDP+ USER'S MANUAL - CHQUS

1.4 DIAGNOSTIC 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 (CHQUS).

2.1 COMMANDS

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

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

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

2.2 SWITCHES

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

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:DDDDD	EXECUTE DDDDD PASSES (DDDDD = 1 TO 64000)
/FLAGS:FLGS	SET SPECIFIED FLAGS. FLAGS ARE DESCRIBED IN SECTION 2.3.
/EOP:DDDDD	REPORT END OF PASS MESSAGE AFTER EVERY DDDDD PASSES ONLY. (DDDDD = 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).

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

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

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

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

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

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

" ! CUSTOMER DATA WILL BE OVERWRITTEN !

CONTINUE (L) ? "

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

THE USER MAY OP1 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)(SEL)	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 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 COMPLETETION OF THE MODIFY OPERARTION, THE USER WILL BE PROMPTED WITH THE FOLLOWING QUESTION;

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

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

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

PARTICULAR DISK ADDRESS;

EXAMPLE 1)

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

CYLINDER ADDRESS (D)  0 ? 598<CR>

TRACK ADDRESS (D)  0 ? 9<CR>

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

SECTOR ADDRESS (D) ? 7<CR>

WORD IN SECTOR (D) ? 150<CR>

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

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

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

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

EXAMPLE 2)

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

CYLINDER ADDRESS (D)  598 ? <CR>

TRACK ADDRESS (D)  9 ? <CR>

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

WORDS FROM INDEX (D) ? 118<CR>

CHANGED TD,
WORD#1  ( CYL)  WORD#2  (TRK)(SEC)  WORD#3  WORD#4  WORD#5  WORD#6
07112€  (598.)  004777  ( 9.)(377)    118.   4891.  140000  140000

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

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

EXAMPLE 3)

M1


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

CYLINDER ADDRESS (D)  598 ? <CR>

TRACK ADDRESS (D)  9 ? <CR>

CURRENT TD,
WORD#1  ( CYL)  WORD#2  (TRK)(SEC)  WORD#3  WORD#4  WORD#5  WORD#6
031126  (598.)  004777  ( 9.)(377)    118.   4891.  140000  140000

TYPE <CR> TO INPUT (0=140000) DATA;
 TD WORD#3 (D)  0 ? 118<CR>

 TD WORD#4 (D)  0 ? 4891<CR>

 TD WORD#5 (D)  0 ? 108<CR>

 TD WORD#6 (D)  0 ? <CR>

CHANGED TD,
WORD#1  ( CYL)  WORD#2  (TRK)(SEC)  WORD#3  WORD#4  WORD#5  WORD#6
071126  (598.)  004777  ( 9.)(377)    118.   4891.    108.  140000

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

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

EXAMPLE 4)

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

    CYLINDER ADDRESS (D)  598 ? 0<CR>

    TRACK ADDRESS (D)  9 ? 6<CR>

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

    SECTOR ADDRESS (D) ? 28<CR>

    WORD IN SECTOR (D) ? (ANY VALID NUMBER)<CR>

    WARNING ** CYL   0., TRK  6. HAS 4 DEFECTS IN TRACK DESCRIPTOR!
               BAD SECTOR FILE WILL BE UPDATED

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

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

EXAMPLE 5)

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

    CYLINDER ADDRESS (D)  0 ? <CR>

    TRACK ADDRESS (D)  6 ? <CR>

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

    WORDS FROM INDEX (D) ? (ANY VALID NUMBER)<CR>

    WARNING ** CYL   0., TRK  6. HAS 4 DEFECTS IN TRACK DESCRIPTOR!
               INPUT DATA REJECTED

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

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

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

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

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

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

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

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

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

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

SKIP DEFECT DISPLACEMENTS.

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

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

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

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

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

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

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

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

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

SKIP DEFECT DISPLACEMENTS.

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

2.6 EXTENDED P-TABLE DIALOGUE

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

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

```
# UNITS (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 (O) 1 ?  0<CR>

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

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

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

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

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

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

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

UNIT 3
CSR ADDRESS (O) ?  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 (D) ?  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 -

F2

THIS MESSAGE IS GENERATED WHEN A COMPOSITE ERRROR 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 PEFORMED.

****

- DRIVE OFFLINE -

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

****

- UNEXPECTED ATTN OCCURRED -

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

****

- WRITE CHECK ERROR SET -

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

****

- FORMAT VERIFY ERROR -

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

****

- DRY DIDN'T RESET WHEN EXPECTED -

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

****

- TRANSFER ERROR SET

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

****

- SPECIAL CONDITION SET -

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

****

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

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

****

- COULD NOT SUPPLY READABLE TD THRU RELOCATION -

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

****

- UNRECOVERABLE ERROR DURING PACK SCAN -

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

****

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

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

****

- NON-EXISTENT DRIVE -

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

**********************************************************************

- DRIVE NOT AN RP07 -

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

**********************************************************************


3.3 SPECIFIC WARNING MESSAGES

**********************************************************************

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

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

**********************************************************************

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

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

**********************************************************************

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

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

**********************************************************************

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

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

**********************************************************************
```

- WARNING ** BAD SECTOR FILE BUFFER IS FULL, NO MORE ENTRIES ALLOWED -

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

****

- WARNING ** BAD SECTOR FILE DATA IS CORRUPT, IT WILL BE INITIALIZED

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

****

- WARNING ** THERE ARE XX. TRACK(S) WITH 4 DEFECTS -

THIS MESSAGE IS GENERATED AT THE END OF A PASS, TO REMINDER THE USER THAT SOME NUMBER OF TRACKS HAD FOUR DEFECTS. WHERE, XX IS THE NUMBER OF TRACKS HAVING FOUR DEFECTS. (DECIMAL)

****

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 DEVICE INFORMATION TABLES

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

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

6.0 PROGRAM PROCESSES

6.1 FORMAT PROCESS

A FORMAT PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

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

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

3. VERIFY TRACK FORMAT, WITH WRITE CHECK (IF ENABLED)

4. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.2 VERIFY PROCESS

A VERIFY PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

1. READ TRACK DESCRIPTOR

2. CONSTRUCT MAP OF EXPECTED FORMAT IN MEMORY BASED ON CONTENTS OF TRACK DESCRIPTOR JUST READ & THE MFG BAD SECTOR FILE INFO.

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

4. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.3 SCAN PROCESS

A SCAN PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

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

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

2. ANY PERSISTENT ERRORS WILL CAUSE THE TRACK DESCRIPTOR TO BE READ AND APPENDED IN MEMORY.

3. WRITE TRACK DESCRIPTOR (ACCORDING TO THE BUFFER CREATED BY THE PACK SCAN OPERATION. IF ENABLED)

4. FORMAT TRACK (WRITE TRACK HEADERS & DATA ACCORDING TO THE DEFECT INFO IN THE TRACK DESCRIPTOR & MFG BAD SECTOR FILE INFO)

5. VERIFY TRACK FORMAT, WITH WRITE CHECK

6. CREATE(USR)-RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

6.4 MODIFY PROCESS

A MODIFY PROCESS WILL CONSIST OF THE FOLLOWING PROGRAM OPERATIONS;

1. READ TRACK DESCRIPTOR

2. MODIFY TRACK DESCRIPTOR

3. WRITE TRACK DESCRIPTOR

4. FORMAT TRACK (WRITE TRACK HEADERS & DATA ACCORDING TO THE DEFECT INFO IN THE TRACK DESCRIPTOR & MFG BAD SECTOR FILE INFO)

5. VERIFY TRACK FORMAT, WITH WRITE CHECK

6. CREATE(USR) RESTORE(MFG) DEC STD 144 BAD SECTOR FILE.

HISTORY

```
.REM @

        VERSION (CZRJK A 0)

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

        VERSION (CZRJK-B 0)

            1.  CHANGED  THE  ROUTINES  THAT DETERMINE  WHEN TO  MOVE THE TRACK
                DESCRIPTOR AND/OR HEADER 0.  THE CHANGE WAS NEEDED BECAUSE OF A
                PARAMETER  LIMIT  DIFFERENCE  BETWEEN   THIS  PROGRAM  AND  THE
                MANUFACTURING SCANNER PROGRAM.

            2.  CHANGED  ROUTINES  'TDSCAN ,  'HDSCAN'  AND  'DASCAN' TO DO 10.
                ITERATIONS  AFTER DETECTING  A  MEDIA RELATED ERROR DURING SCAN
                MODE.

            3.  CHANGED  THE  HEADER SCAN  ROUTINE 'HDSCAN',  TO LOOK FOR 'OPI'
                ERRORS DURING SCAN MODE.

            4.  CORRECTED A SHIFT RIGHT PROBLEM IN ROUTINE 'TOLER'.

            5.  MODIFIED THE  PROGRAM  TO FORMAT  THE  MF BIT  CORRECTLY INTO A
                HEADER THAT HAS BEEN FOUND IN THE MFG BAD SECTOR FILE.

            6.  MODIFIED  THE  PROGRAM TO  ALLOW  THE USER TO FORMAT,  SCAN  OR
                MODIFY TD'S  ON THE FE CYLINDER,  WITHOUT HAVING TO  CHANGE THE
                MIN/MAX CYLINDER LIMITS IN THE DRIVE PARAMETERS.
@
```

M?


```
HISTORY

      1                                 ;*LAST REVISION 25-MAY-83
      2
    136                                 .TITLE  CZRJKB0 RP07 FMTR/SCANNER
    138                                 .SBTTL  PROGRAM HEADER
    164
    166 000000                                  .ENABL  AMA,ABS
    167         002000                          .       =       2000
    169
    171                                 ;++
    172                                 ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
    173                                 ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
    174                                 ;
    175
    177
    194
    198 002000                          L$NAME::                ;DIAGNOSTIC NAME
        002000      103                         .ASCII /C/
        002001      132                         .ASCII /Z/
        002002      122                         .ASCII /R/
        002003      112                         .ASCII /J/
        002004      113                         .ASCII /K/
        002005      000                         .BYTE   0
        002006      000                         .BYTE   0
        002007      000                         .BYTE   0
        002010                          L$REV::                 ;REVISION LEVEL
        002010      102                         .ASCII  /B/
        002011                          L$DEPO::                ;0
        002011      060                         .ASCII  /0/
        002012                          L$UNIT::                ;NUMBER OF UNITS
        002012  000001                          .WORD   T$PTHV
        002014                          L$TIML::                ;LONGEST TEST TIME
        002014  001000                          .WORD   1000
        002016                          L$HPCP::                ;POINTER TO H.W. QUES.
        002016  031132                          .WORD   L$HARD
        002020                          L$SPCP::                ;POINTER TO S.W. QUES.
        002020  000000                          .WORD   0
        002022                          L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
        002022  002130                          .WORD   L$HW
        002024                          L$SPTP::                ;PTR. TO S.W. PTABLE
        002024  000000                          .WORD   0
        002026                          L$LADP::                ;DIAG. END ADDRESS
        002026  120100                          .WORD   L$LAST
        002030                          L$STA::                 ;RESERVED FOR APT STATS
        002030  000000                          .WORD   0
        002032                          L$CO::
        002032  000000                          .WORD   0
        002034                          L$DTYP::                ;DIAGNOSTIC TYPE
        002034  000000                          .WORD   0
        002036                          L$APT::                 ;APT EXPANSION
        002036  000000                          .WORD   0
        002040                          L$DTP::                 ;PTR. TO DISPATCH TABLE
        002040  002124                          .WORD   L$DISPATCH
        002042                          L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
        002042  000000                          .WORD   0
        002044                          L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
        002044  000000                          .WORD   0
        002046                          L$EXP1::                ;EXPANSION WORD
```

PROGRAM HEADER

```
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002614                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  000000                  .WORD   0
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  000000                  .WORD   0
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  000000                  .WORD   0
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002622                  .WORD   L$DESC
002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                 ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                 ;PTR. TO INIT CODE
002104  026516                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  027552                  .WORD   L$CLEAN
002110                  L$ACP::                 ;PTR. TO AUTO CODE
002110  027550                  .WORD   L$AUTO
002112                  L$PRT::                 ;PTR. TO PROTECT TABLE
002112  026510                  .WORD   L$PROT
002114                  L$TEST::                ;TEST NUMBER
002114  000000                  .WORD   0
002116                  L$DLY::                 ;DELAY COUNT
002116  000000                  .WORD   0
002120                  L$HIME::                ;PTR. TO HIGH MEM
002120  000000                  .WORD   0
200
```

```
                                .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                          .WORD   1
002124                          L$DISPATCH::
002124  030074                          .WORD   T1
```

```
                                .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                          .WORD   L10000-L$HW/2
002130                          L$HW::
002130                          DFPTBL::
002130  176700                          .WORD   176700          ;RPCS1 BASE REGISTER ADDRESS
002132  000254                          .WORD   254             ;VECTOR ADDRESS
002134  000240                          .WORD   240             ;BR LEVEL 5 DEVICE
002136  000000                          .WORD   0               ;DRIVE NUMBER

002140                          L10000:
```

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

GLOBAL EQUATES SECTION

```
                                .SBTTL  GLOBAL EQUATES SECTION


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


                                ;
                                ; BIT DEFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.                             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                             ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.                             ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.                             ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.                             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
000340                          PRI07== 340
000300                          PRI06== 300
000240                          PRI05== 240
000200                          PRI04== 200
000140                          PRI03== 140
000100                          PRI02== 100
```

GLOBAL EQUATES SECTION

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

```
                                .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)
```

```
                                ;DRIVE STATUS REGISTER (RPDS <#01>)

000001          OM      == 1            ;OFFSET MODE (BIT #0)
000002          EWN     == 2            ;EARLY WARNING (BIT #1)
000004          ILEV    == 4            ;INTERLEAVING AVAILABLE (BIT #2)
000100          VV      == 100          ;VOLUME VALID (BIT #6)
000200          DRY     == 200          ;DATA READY (BIT #7)
000400          DPR     == 400          ;DRIVE PRESENT (BIT #8)
001000          PGM     == 1000         ;PROGRAMABLE (BIT #9)
002000          LBT     == 2000         ;LAST BLOCK TRANSFERRED (BIT #10)
004000          WRL     == 4000         ;WRITE LOCKED (BIT #11)
010000          MOL     == 10000        ;MEDIUM ON LINE (BIT #12)
020000          PIP     == 20000        ;POSITIONER IN PROGRESS (BIT #13)
040000          ERR     == 40000        ;COMPOSITE ERROR (BIT #14)
100000          ATA     == 100000       ;ATTENTION ACTIVE (BIT #15)


                                ;ERROR REGISTER #1 (RPER1 <#02>)

000001          ILF     == 1            ;ILLEGAL FUNCTION (BIT #0)
000002          ILR     == 2            ;ILLEGAL REGISTER (BIT #1)
000004          RMR     == 4            ;REGISTER MODIFICATION REFUSED (BIT #2)
000010          PAR     == 10           ;PARITY ERROR (BIT #3)
000020          FER     == 20           ;FORMAT ERROR (BIT #4)
000040          WCF     == 40           ;WRITE CLOCK FAIL (BIT #5)
000100          ECH     == 100          ;ECC HARD ERROR (BIT #6)
000200          HCE     == 200          ;HEADER COMPARE ERROR (BIT #7)
000400          HCRC    == 400          ;HEADER CRC ERROR (BIT #8)
001000          AOE     == 1000         ;ADDRESS OVERFLOW ERROR (BIT #9)
002000          IAE     == 2000         ;INVALID ADDRESS ERROR (BIT #10)
004000          WLE     == 4000         ;WRITE LOCK ERROR (BIT #11)
010000          DTE     == 10000        ;DRIVE TIMING ERROR (BIT #12)
020000          OPI     == 20000        ;OPERATION INCOMPLETE (BIT #13)
040000          UNS     == 40000        ;DRIVE UNSAFE (BIT #14)
100000          DCK     == 100000       ;DATA CHECK ERROR (BIT #15)


                                ;DIAGNOSTIC MAINTAINABLILTY REGISTER (RPMR1 <#03>)

100000          DMD     == 100000       ;DIAGNOSTIC MODE (BIT #15)


                                ;ATTENTION SUMMARY PSEUDO REGISTER (RPAS <#04>)

000001          AT0     == 1            ;DEVICE 0 (BIT #0)
000002          AT1     == 2            ;DEVICE 1 (BIT #1)
000004          AT2     == 4            ;DEVICE 2 (BIT #2)
000010          AT3     == 10           ;DEVICE 3 (BIT #3)
000020          AT4     == 20           ;DEVICE 4 (BIT #4)
000040          AT5     == 40           ;DEVICE 5 (BIT #5)
000100          AT6     == 100          ;DEVICE 6 (BIT #6)
000200          AT7     == 200          ;DEVICE 7 (BIT #7)


                                ;DESIRED SECTOR/TRACK ADDRESS REGISTER (RPDA <#05>)
                                ;EACH BIT IS CALLED BY BIT NUMBER
```

```
                                ;DRIVE TYPE REGISTER (RPDT <#06>)

000001                  DRT0    == 1                    ;DRIVE TYPE NUMBER (BIT #0)
000002                  DRT1    == 2                    ;DRIVE TYPE NUMBER (BIT #1)
000004                  DRT2    == 4                    ;DRIVE TYPE NUMBER (BIT #2)
000010                  DRT3    == 10                   ;DRIVE TYPE NUMBER (BIT #3)
000020                  DRT4    == 20                   ;DRIVE TYPE NUMBER (BIT #4)
000040                  DRT5    == 40                   ;DRIVE TYPE NUMBER (BIT #5)
000100                  DRT6    == 100                  ;DRIVE TYPE NUMBER (BIT #6)
000200                  DRT7    == 200                  ;DRIVE TYPE NUMBER (BIT #7)
000400                  DRT8    == 400                  ;DRIVE TYPE NUMBER (BIT #8)
004000                  DRQ     == 4000                 ;DRIVE REQUEST REQUIRED (BIT #11)
020000                  MOH     == 20000                ;MOVING HEAD TYPE DRIVE (BIT #13)
040000                  TAP     == 40000                ;TAPE DRIVE (BIT #14)
100000                  NBA     == 100000               ;NOT BLOCK ADDRESSED (BIT #15)


                                ;LOOK AHEAD REGISTER (RPLA <#07>)

000100                  SC1     == 100                  ;SECTOR COUNT FIELD 1 (BIT #6)
000200                  SC2     == 200                  ;SECTOR COUNT FIELD 2 (BIT #7)
000400                  SC4     == 400                  ;SECTOR COUNT FIELD 4 (BIT #8)
001000                  SC8     == 1000                 ;SECTOR COUNT FIELD 8 (BIT #9)
002000                  SC16    == 2000                 ;SECTOR COUNT FIELD 16 (BIT #10)
004000                  SC32    == 4000                 ;SECTOR COUNT FIELD 32 (BIT #11)
010000                  SC64    == 10000                ;SECTOR COUNT FIELD 64 (BIT #12)


                                ;RP07 SERIAL NUMBER REGISTER (RPSN <#10>)
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;RP07 OFFSET REGISTER (RPOF <#11>)

000200                  OFFDIR  == 200                  ;OFFSET DIRECTION (BIT #7)
002000                  HCI     == 2000                 ;HEADER COMPARE CODE INHIBIT (BIT #10)
004000                  ECI     == 4000                 ;ERROR CORRECTION CODE INHIBIT (BIT #11)
010000                  FMT16   == 10000                ;16 BIT FORMAT (BIT #12)
040000                  MTD     == 40000                ;MOVE TRACK DESCRIPTOR (BIT #14)
100000                  CMOD    == 100000               ;COMMAND MODIFIER (BIT #15)


                                ;RP07 DESIRED CYLINDER ADDRESS (RPDC <#12>)
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;RP07 CURRENT CYLINDER ADDRESS (RPCC <#13>)
                                ;EACH BIT IS CALLED BY BIT NUMBER


                                ;RP07 ERROR REGISTER 3 (RPER3 <#15>)

000002                  SCF     == 2                    ;SYNC CLOCK FAI'URE (BIT #1)
000004                  SBE     == 4                    ;SYNC BYTE ERROR (BIT #2)
000010                  DPE     == 10                   ;DATA PARITY ERROR (BIT #3)
```

```
                        SDF     == 20                   ;SERDES DATA FAILURE (BIT #4)
                        DCU     == 40                   ;DC UNSAFE (BIT #5)
                        IXU     == 100                  ;INDEX UNSAFE (BIT #6)
                        DVC     == 200                  ;DEVICE CHECK (BIT #7)
                        PHF     == 400                  ;8080 PROCESSOR HANDSHAKE FAILURE (BIT #8)
                        LCE     == 1000                 ;LOSS OF CYLINDER ERROR (BIT #9)
                        LBC     == 2000                 ;LOSS OF BIT CLOCK (BIT #10)
                        DSE     == 20000                ;DEFECT SKIP ERROR (BIT #13)
                        SKI     == 40000                ;SEEK INCOMPLETE (BIT #14)
                        BSE     == 100000               ;BAD SECTOR ERROR (BIT #15)


                        ;RP07 ERROR REGISTER #2 (RPER2 <#14>)

                        WRU     == 400                  ;WRITE READY UNSAFE (BIT #8)
                        WOR     == 1000                 ;WRITE OVERRUN (BIT #9)
                        RWU1    == 2000                 ;READ/WRITE UNSAFE #1 (BIT #10)
                        RWU2    == 4000                 ;READ/WRITE UNSAFE #2 (BIT #11)
                        RWU3    == 10000                ;READ/WRITE UNSAFE #3 (BIT #12)
                        CPU     == 20000                ;CPU UNSAFE (BIT #13)
                        CPE     == 40000                ;CROM PARITY ERROR (BIT #14)
                        PGE     == 100000               ;PROGRAMING ERROR


                        ;ECC POSITION REGISTER (RPEC1 <#16>)
                        ;EACH BIT IS DEFINED BY BIT NUMBER


                        ;ECC PATTERN REGISTER (RPEC2 <#17>)
                        ;EACH BIT IS DEFINED BY BIT NUMBER

                        .SBTTL  RP07 DRIVER COMMANDS

                        NOOP    == 1                    ;NO OPERATION
                        SEEK    == 5                    ;SEEK
                        RECAL   == 7                    ;RECALIBRATE
                        DRVCLR  == 11                   ;DRIVE CLEAR
                        RELSE   == 13                   ;RELEASE
                        READIN  == 21                   ;READ IN PRESET
                        SEARCH  == 31                   ;SEARCH
                        DIAG    == 35                   ;DIAGNOSTIC MODE
                        WCKD    == 51                   ;WRITE CHECK DATA
                        WCKHD   == 53                   ;WRITE CHECK HEADER & DATA
                        WRTDAT  == 61                   ;WRITE DATA
                        FMTRK   == 63                   ;FORMAT TRACK
                        WRTTD   == 65                   ;WRITE TRACK DESCRIPTOR
                        RDDAT   == 71                   ;READ DATA
                        RDHD    == 73                   ;READ HEADER & DATA
                        RDTD    == 75                   ;READ TRACK DESCRIPTOR

                        ;SOME TRACK FORMAT EQUATES FOR 16 BIT MODE (IN WORDS)

                        G1      == 64.                  ;SIZE OF GAP 1
                        TD      == 8.                   ;SIZE OF TD
                        G2      == 37.                  ;SIZE OF GAP 2
                        HDR0    == 8.                   ;SIZE OF HEADER 0
                        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
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

            002142              BGNPAT  ==.
002142  000000                          .WORD   000000          ;PATTERN #1 (ALL 0'S)
002144  000000                          .WORD   000000
002146  177777                          .WORD   177777          ;PATTERN #2 (ALL 1'S)
002150  177777                          .WORD   177777
002152  030221                          .WORD   030221          ;PATTERN #3 (WORST CASE)
002154  030221                          .WORD   030221
            002156              ENDPAT  ==.

002156  000000                  ANYWHR:: .WORD  0               ;IF = 1, ALLOW ACCESS TO ANYWHERE ON MEDIA
                                                                ;IF = 0, ALLOW ACCESS TO FE CYLINDER 1 ONLY
002160  000000                  OPTION:: .WORD  0               ;OPTIONS; 0=FORMAT, 1=VERIFY, 2=SCAN, 3=MODIFY,
                                                                ;         4=LIST, 5=WRITE FE-2 AND 6=HELP
002162  000000                  FORMT:: .WORD   0               ;FORMAT OPTION; ENABLED= 1, DISABLED= 0
002164  000000                  VRIFY:: .WORD   0               ;VERIFY OPTION; ENABLED= 1, DISABLED= 0
002166  000000                  SCANR:: .WORD   0               ;SCAN OPTION; ENABLED= 1, DISABLED= 0
002170  000000                  MODTD:: .WORD   0               ;MODIFY OPTION; ENABLED= 1, DISABLED= 0
002172  000000                  LIST::  .WORD   0               ;LIST OPTION; ENABLED= 1, DISABLED= 0
002174  000000                  WRTFE2:: .WORD  0               ;WRITE 2ND FE CYL OPTION; ENABLED= 1, DISABLED= 0

002176  000000                  NOWRCK:: .WORD  0               ;USED TO INHIBIT WRITE CHECK COMMAND
002200  000000                  ENWTTD:: .WORD  0               ;USED TO ENABLE WRITE TD OPTION AFTER PACK SCAN
002202  000000                  MODBY:: .WORD   0               ;MODIFY TD BY; 0=WORDS, 1=INDEX, 2=CHANGE
002204  000000                  LISHDR:: .WORD  0               ;USED TO LIST HEADER INFO IN DEFECT SECTORS
002206  000000                  DRVPAR:: .WORD  0               ;WHEN EQ TO 1 CHANGE DRIVE PARAMETERS

002210  000000                  MINCYL:: .WORD  0               ;USED TO DETERMINE MIN CYLINDER OF OPERATION
002212  001166                  MAXCYL:: .WORD  630.            ;USED TO DETERMINE MAX CYLINDER OF OPERATION
002214  001166                  TMPCYL:: .WORD  630.            ;TEMPORARY STORAGE FOR MAXIMUM CYLINDER VALUE
002216  000000                  MINTRK:: .WORD  0               ;USED TO DETERMINE MIN TRACK OF OPERATION
002220  000037                  MAXTRK:: .WORD  31.             ;USED TO DETERMINE MAX TRACK OF OPERATION

002222  001165                  LSCYL:: .WORD   629.            ;LAST USER CYLINDER ADDRESS
002224  000037                  LSTRK:: .WORD   31.             ;LAST TRACK ADDRESS
002226  000005                  ERRMAX:: .WORD  5               ;USED TO CONTROL MAX ERRORS
002230  000310                  ENDPTR:: .WORD  50.*4           ;USED TO CREATE BUFFER BOUNDARIES
002232  000000                  ENDTAB:: .WORD  0               ;USED TO MARK BUFFER BOUNDARIES
002234  000000                  BUSADR:: .WORD  0               ;CONTAINS BUS ADDR FOR DRIVER MODULE
002236  000000                  OFFSET:: .WORD  0               ;CONTENTS OF OFFSET REGISTER FOR DRIVER MODULE
002240  032464                  LASLOC:: .WORD   DEFBUF         ;USED TO LINK BUFFER TO BUFFER TRANSFERS

002242      000                 FIRPAS:: .BYTE  0               ;USED TO CONTROL SCANNER ITERATIONS
002243      000                 SCANIT:: .BYTE  0               ;# OF SCAN ITERATIONS DURING A SCAN OPERATION
002244      000                 INTLEV:: .BYTE  0               ;INTERLEAVE MODE INDICATOR
002245      000                 SAMSEC:: .BYTE  0               ;# DEFECTS IN CURRENT SECTOR
002246      000                 NEWCNT:: .BYTE  0               ;USED TO TRACK NEWLY FOUND DEFECTS
002247      000                 CONFLG:: .BYTE  0               ;SET IF PROGRAM IS DOING CONTINUE COMMAND
                                        .EVEN
002250  000000                  TEMPA::  .WORD  0               ;USED FOR SOFTWARE CALCULATIONS
002252  000000                  SUPRSS:: .WORD  0               ;CONTROL USED TO SUPRESS ERROR MESSAGES
```

GLOBAL DATA SECTION

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

N3


```
                                CONTLT:: .WORD  0               ;ADDRESS OF CONTROL 'T' SERVICE BUFFER
                                TKWCNT:: .WORD  6*50.           ;WORDS FOR TRACK FORMAT (6 WORDS X 50. SECTORS)
                                TEMP1:: .WORD   0               ;USED TO INDENTIFY SCRATCHES
                                TEMP2:: .WORD   0               ;USED TO MEASURE SCRATCH LENGTH
                                HICYL:: .WORD   0               ;USED TO MAP THE HI ADDRESS OF A SCRATCH
                                LOCYL:: .WORD   0               ;USED TO MAP THE LO ADDRESS OF A SCRATCH
                                NEGWRD:: .WORD  0               ;NEGATED WORD COUNT FOR DRIVER
                                RELWRD:: .WORD  0               ;CONTAINS # OF WORDS TO DEFECT, RELATIVE TO INDEX
                                SECADD:: .WORD  0               ;SECTOR ADDRESS OF NEW TD DEFECT
                                FUNCTN:: .WORD  0               ;FUNCTION COMMAND FOR RP07 DRIVER MODULE
                                DEFSEC:: .WORD  0               ;CONTAINS # OF DEFECTS IN A SECTOR
                                DEFTRK:: .WORD  0               ;CONTAINS # OF DEFECTS ON A TRACK
                                MINSEG:: .WORD  0               ;MINIMUM VALUE FOR A DATA SEGMENT
                                MAXSEG:: .WORD  0               ;MAXIMUM VALUE FOR A DATA SEGMENT
                                MORETD:: .WORD  0               ;IF EQ 1, THEN MORE TD'S TO MODIFY

                                UNIT::  .WORD   0               ;USED TO SELECT A UNIT NUMBER
                                RPADR:: .WORD   176700          ;CONTAINS RPCS1 BASE ADDRESS
                                RPVEC:: .WORD   254.5*32.       ;CONTAINS VECTOR ADDRESS & BR LEVEL
                                RHEXT:: .WORD   50              ;CONTAINS RH70 OFFSET TO RPBAE
                                RHTYPE::.WORD   0               ;CONTAINS RHXX TYPE; RH11= 0, RH70= 1
                                DRVNO:: .WORD   0               ;DRIVE NUMBER
                                DRVSN:: .WORD   0               ;STORAGE FOR EACH S/N DIGIT

                                RPCS1:: .WORD   176700          ;BASE ADDRESS USED FOR THE DRIVE
                                RPWC::  .WORD   176702          ;WORD COUNT REGISTER
                                RPBA::  .WORD   176704          ;BYTE ADDRESS REGISTER
                                RPDA::  .WORD   176706          ;DESIRED SECTOR/TRACK ADDRESS
                                RPCS2:: .WORD   176710          ;RP07 STATUS REGISTER
                                RPDS::  .WORD   176712          ;RP07 DRIVE STATUS
                                RPER1:: .WORD   176714          ;RP07 ERROR REGISTER #1
                                RPAS::  .WORD   176716          ;RP07 ATTENTION SUMMARY PSEUDO REGISTER
                                RPLA::  .WORD   176720          ;RP07 LOOK AHEAD REGISTER
                                RPDB::  .WORD   176722          ;RP07 DATA BUFFER
                                RPMR1:: .WORD   176724          ;RP07 MAINTENANCE REGISTER #1
                                RPDT::  .WORD   176726          ;DRIVE TYPE REGISTER
                                RPSN::  .WORD   176730          ;RP07 SERIAL NUMBER
                                RPOF::  .WORD   176732          ;RP07 OFFSET REGISTER
                                RPDC::  .WORD   176734          ;RP07 DESIRED CYLINDER
                                RPCC::  .WORD   176736          ;RP07 CURRENT CYLINDER
                                RPER2:: .WORD   176740          ;RP07 ERROR REGISTER #2
                                RPER3:: .WORD   176742          ;RP07 ERROR REGISTER #3
                                RPEC1:: .WORD   176744          ;RP07 ERROR-POSITION
                                RPEC2:: .WORD   176746          ;RP07 ERROR PATTERN
                                RPBAE:: .WORD   176750          ;RH70 REGISTER
                                RPCS3:: .WORD   176752          ;RH70 REGISTER

                                ; STORAGE FOR DEVICE REGISTERS
                                ;
                                REG::   .BLKW   22.             ;BUFFER TO SAVE REGISTERS AFTER AN ERROR

                                DELTA:: .BLKW   4               ;BUFFER USED TO STORE DELTA ADJUSTMENTS
                                CMDQUE:: .BLKW  6               ;USED FOR A COMMAND SEQUENCE QUEUE
```

```
                                        .SBTTL  GLOBAL TEXT SECTION

                                        ;++
                                        ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                        ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                        ; MORE THAN ONE TEST.
                                        ;

                                        ;NAMES OF DEVICES SUPPORTED BY PROGRAM
                                        ;
002614                                  L$DVTYP::
002614  122     120     060                     .ASCIZ  /RP07/
                                                .EVEN

                                        ; TEST DESCRIPTION
                                        ;
002622                                  L$DESC::
002622  122     120     060                     .ASCIZ  /RP07 FORMAT VERIFY-SCANNER/
                                                .EVEN

                                        ; FORMAT STATEMENTS USED IN PRINT CALLS
                                        ;
002656  045     116     000     CRLF::  .ASCIZ /%N/
002661  045     124     000     FRMTT:: .ASCIZ /%T/
002664  045     116     045     FRMT00:: .ASCIZ /%N%ADRIVE   RPCS1   RPWC    RPBA    RPDA    RPCS2   RPDS/
002755  045     116     045     FRMT01:: .ASCIZ /%N%06%A  %06%A  %06%A  %06%A  %06%A  %06/
003035  045     116     045     FRMT02:: .ASCIZ /%N%ARPER1   RPAS    RPLA    RPDB    RPMR1   RPDT    RPSN/
003126  045     116     045     FRMT03:: .ASCIZ /%N%06%A  %06%A  %06%A  %06%A  %06%A  %06%A  %06/
003206  045     116     045     FRMT04:: .ASCIZ /%N%ARPOF    RPDC    RPCC    RPER2   RPER3   RPEC1   RPEC2/
003300  045     116     045     FRMT05:: .ASCIZ /%N%06%A  %06%A  %06%A  %06%A  %06%A  %06%A  %06%N/
003362  045     101     122     FRMT06:: .ASCIZ /%ARPBAE   RPCS3/
003402  045     116     045     FRMT07:: .ASCIZ /%N%06%A  %06%N/

003421  045     116     045     FRMT10:: .ASCIZ /%N%ADRIVE %01/
003437  045     116     045     FRMT11:: .ASCIZ /%N%ADONE.  RETRIES MADE= %D5%A., ERRORS DETECTED= %D5%A.%N/
003532  045     101     124     FRMT12:: .ASCIZ /%ATRACKS WITH %D2%A DEFECTS= %D5%A.%N/
003600  045     116     045     FRMT13:: .ASCIZ /%N%ATOTAL DEFECTS FOUND= %D5%A./
003640  045     104     065     FRMT14:: .ASCIZ /%D5%A.  /
003651  045     117     066     FRMT15:: .ASCIZ /%06%A  /
003661  045     116     045     FRMT16:: .ASCIZ /%N%ADRIVE   WORD#1  WORD#2  WORD#3  WORD#4  WORD#5  WORD#6/
003754  045     116     045     FRMT17:: .ASCIZ /%N%06%A  %06%A  %06%A  %06%A  %06%A  %06%A  %06/

004034  045     116     045     FRMT20:: .ASCIZ /%N%ACYL:%D3%A.  TRK:%D2%A. /
004070  045     101     120     FRMT22:: .ASCIZ /%APRSNT FUNCT: %T/
004112  045     124     045     FRMT23:: .ASCIZ /%T%A  %06%A  %06/
004133  045     116     045     FRMT24:: .ASCIZ /%N%ATYPE <CR> TO INPUT (0-140000) DATA;/

004203  045     101     054     FRMT30:: .ASCIZ /%A, NON-INTERLEAVED%N/
004231  045     101     054     FRMT31:: .ASCIZ /%A, INTERLEAVED%N/
004253  045     116     045     FRMT32:: .ASCIZ /%N%ANEW DEFECT(S) DURING SCAN ITERATION %D3%A., TD(S) FOLLOW:/
004351  045     116     045     FRMT33:: .ASCIZ /%N%AEXPCTD:%06%A  RECVED:%06/
004406  045     116     045     FRMT35:: .ASCIZ /%N%AWORD#1 ( cyl)  WORD#2 (trk)(sec)  WORD#3  WORD#4  WORD#5  WORD#6%N/
004515  045     117     066     FRMT36:: .ASCIZ /%06%A (%D3%A.)  %06%A (%D2%A.)(%03%A)  /
004565  045     101     040     FRMT37:: .ASCIZ /%A                %06%A      (%D2%A.)  /
```

GLOBAL TEXT SECTION

```
 71
 72 004634      045     116     045  FRMT50::.ASCII  /%N%AAVAILABLE OPTIONS FOLLOW:/
 73 004671      045     116     045          .ASCII  /%N%A 0 -Format/
 74 004707      045     116     045          .ASCII  /%N%A 1 -Verify/
 75 004725      045     116     045          .ASCII  /%N%A 2 -Scan/
 76 004742      045     116     045  FRMT51::.ASCII  /%N%A 3 -List/
 77 004756      045     116     045          .ASCII  /%N%A 4 -Modify/
 78 004774      045     116     045          .ASCIZ  /%N%A 5 -Write FE-2%N/
 79
 80                                  .SBTTL  WARNING MESSAGES
 81
 82 005021      045     123     061  FRMT60:: .ASCIZ /%S11%AInput data rejected%N/
 83 005055      045     116     045  FRMT61:: .ASCIZ /%N%T%ACYL %D3%A., TRK %D2%A. has 4 defects in track descriptor!%N/
 84 005157      045     116     045  FRMT62:: .ASCIZ /%N%T%AThere are %D5%A. track(s) with 4 defects!%N/
 85 005241      045     116     045  FRMT63:: .ASCIZ /%N%T%ABad sector file buffer is full, no more entries allowed!%N/
 86 005342      045     116     045  FRMT64:: .ASCIZ /%N%T%ABad sector file data is corrupt, it will be initialized!%N/
 87 005443      045     123     061  FRMT65:: .ASCIZ /%S11%AAnother defect was found %D5%A. words from Index%N/
 88 005534      045     123     061  FRMT66:: .ASCIZ /%S11%ABad sector file will be updated%N/
 89
 90                                  .SBTTL  GLOBAL ASCII MESSAGE SECTION
 91
 92 005604      127     122     111  MSWRTD:: .ASCIZ /WRITE TD/
 93 005615      122     105     101  MSRDTD:: .ASCIZ /READ TD/
 94 005625      127     122     124  WRDAT:: .ASCIZ  /WRT DATA/
 95 005636      127     122     124  WCKDAT:: .ASCIZ /WRT CHK DATA/
 96 005653      127     122     124  WCKHDR:: .ASCIZ /WRT CHK HEADER/
 97 005672      122     105     103  RECALI:: .ASCIZ /RECAL/
 98 005700      106     117     122  FORMAT:: .ASCIZ /FORMAT TRK/
 99 005713      122     105     101  RHDATA:: .ASCIZ /READ HEADER/
100 005727      124     105     123  MOLINE:: .ASCIZ /TEST MOL- 1/
101 005743      124     105     123  WLOCK:: .ASCIZ  /TEST WRL- 0/
102 005757      124     105     123  DRVRDY:: .ASCIZ /TEST RPDS: BIT7- 1/
103 006002      015     012     105  EXPTD:: .ASCIZ  <CR><LF>/EXPCTD: /
104 006015      015     012     122  RCVED:: .ASCIZ  <CR><LF>/RECVED: /
105 006030      015     012     103  CURENT:: .ASCIZ <CR><LF>/CURRENT TD./
106 006046      015     012     124  TDDEF:: .ASCIZ  <CR><LF>/TD(S) WITH DEFECTS FOLLOW:/
107 006103      015     012     124  ANDHDR:: .ASCIZ <CR><LF>/TD(S) WITH DEFECTS (INCLUDING HEADER INFO) FOLLOW:/
108 006170      015     012     103  CHANGE:: .ASCIZ <CR><LF>/CHANGED TD./
109 006206      127     101     122  WARN:: .ASCIZ   /WARNING ** /<BELL><BELL>
110
111 006224      103     110     101  MESG18:: .ASCIZ /CHANGE DRIVE PARAMETERS/
112 006254      105     116     124  MESG7:: .ASCIZ  /ENTER OPTION (6-HELP)/
113 006302      115     111     116  MESG13:: .ASCIZ /MIN TRK/
114 006312      115     101     130  MESG14:: .ASCIZ /MAX TRK/
115 006322      115     111     116  MESG15:: .ASCIZ /MIN CYL/
116 006332      115     101     130  MESG16:: .ASCIZ /MAX CYL/
118 006342      104     117     040  WRITMG: .ASCIZ  /DO YOU WANT TO WRITE ANYWHERE ON MEDIA/
119 006411      007     011     041  WRSAFM: .ASCII  <BELL>/ ! CUSTOMER DATA WILL BE OVERWRITTEN !/<CR><LF>
120 006462      007     011     055          .ASCII  <BELL>/ ----------------------------------/<CR><LF>
121 006533      103     117     116          .ASCIZ  /CONTINUE/
123 006544      111     116     110  MESG17:: .ASCIZ /INHIBIT WRITE CHECK/
124 006570      115     117     104  MESG22:: .ASCIZ /MODIFY BY (0 -Words, 1 -Index, 2 -Change)/
125 006642      104     117     040  MESG19:: .ASCIZ /DO YOU WANT TO RE-WRITE TD(S) WITH NEW DEFECTS/
126 006721      114     111     123  MESG23:: .ASCIZ /LIST HEADER INFO IN DEFECT SECTOR(S)/
127
128 006766      124     104     040  EM1::   .ASCIZ  /TD ADDRESS INCORRECT/
129 007013      124     104     040  EM2::   .ASCIZ  /TD FORMAT INCORRECT/
```

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

```
                                .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..
                                ;--

007722                          ERR0::
007722  004737  011046                  JSR     PC,TYPLOC       ;REPORT THE ADDRESS OF THE FAILURE
007726  004737  011132                  JSR     PC,DMPREG       ;DUMP THE ERROR REGISTERS
                                                                ;CR-LF
007732  012746  002656                  MOV     #CRLF,-(SP)
007736  012746  000001                  MOV     #1,-(SP)
007742  010600                          MOV     SP,R0
007744  104414                          TRAP    C$PNTB
007746  062706  000004                  ADD     #4,SP
007752                          L10002:
007752  104423                          TRAP    C$MSG

007754                          ERR1::
007754  004737  011046                  JSR     PC,TYPLOC       ;REPORT THE ADDRESS OF THE FAILURE
                                                                ;PRINT 'EXPCTD: XXXXXX XXXXXX'
007760  013746  031432                  MOV     TDCPY2,-(SP)
007764  013746  031430                  MOV     TDCPY1,-(SP)
007770  012746  006002                  MOV     #EXPTD,-(SP)
007774  012746  004112                  MOV     #FRMT23,-(SP)
010000  012746  000004                  MOV     #4,-(SP)
010004  010600                          MOV     SP,R0
010006  104414                          TRAP    C$PNTB
010010  062706  000012                  ADD     #12,SP
                                                                ;PRINT 'RECVED: XXXXXX XXXXXX'
010014  013746  031416                  MOV     TDWRD2,-(SP)
010020  013746  031414                  MOV     TDWRD1,-(SP)
010024  012746  006015                  MOV     #RCVED,-(SP)
010030  012746  004112                  MOV     #FRMT23,-(SP)
010034  012746  000004                  MOV     #4,-(SP)
010040  010600                          MOV     SP,R0
010042  104414                          TRAP    C$PNTB
010044  062706  000012                  ADD     #12,SP
010050  004737  011132                  JSR     PC,DMPREG       ;DUMP THE ERROR REGISTERS
                                                                ;CR-LF
010054  012746  002656                  MOV     #CRLF,-(SP)
010060  012746  000001                  MOV     #1,-(SP)
010064  010600                          MOV     SP,R0
010066  104414                          TRAP    C$PNTB
010070  062706  000004                  ADD     #4,SP
010074                          L10003:
010074  104423                          TRAP    C$MSG

010076                          ERR2::
010076  004737  011046                  JSR     PC,TYPLOC       ;REPORT THE ADDRESS OF THE FAILURE
                                                                ;PRINT 'DRIVE WORD#1 WORD#2 WORD#3 WORD#4 WORD#5 WORD#6'
010102  012746  003661                  MOV     #FRMT16,-(SP)
010106  012746  000001                  MOV     #1,-(SP)
010112  010600                          MOV     SP,R0
010114  104414                          TRAP    C$PNTB
```

GLOBAL ERROR REPORT SECTION

```
                010116  062706  000004                  ADD     #4,SP
31              010122  013746  031426                  MOV     TDWRD6,-(SP)
                010126  013746  031424                  MOV     TDWRD5,-(SP)
                010132  013746  031422                  MOV     TDWRD4,-(SP)
                010136  013746  031420                  MOV     TDWRD3,-(SP)
                010142  013746  031416                  MOV     TDWRD2,-(SP)
                010146  013746  031414                  MOV     TDWRD1,-(SP)
                010152  013746  002434                  MOV     CRVNO,-(SP)
                010156  012746  003754                  MOV     #FRMT17,-(SP)
                010162  012746  000010                  MOV     #10,-(SP)
                010166  010600                          MOV     SP,R0
                010170  104414                          TRAP    C$PNTB
                010172  062706  000022                  ADD     #22,SP
32                                                                              ;CR-LF
33              010176  012746  002656                  MOV     #CRLF,-(SP)
                010202  012746  000001                  MOV     #1,-(SP)
                010206  010600                          MOV     SP,R0
                010210  104414                          TRAP    C$PNTB
                010212  062706  000004                  ADD     #4,SP
34              010216                          L10004:
                010216  104423                          TRAP    C$MSG
35
36              010220                          ERR3::
37              010220  004737  011046                  JSR     PC,TYPLOC       ;REPORT THE ADDRESS OF THE FAILURE
38                                                                              ;CR-LF
39              010224  012746  002656                  MOV     #CRLF,-(SP)
                010230  012746  000001                  MOV     #1,-(SP)
                010234  010600                          MOV     SP,R0
                010236  104414                          TRAP    C$PNTB
                010240  062706  000004                  ADD     #4,SP
40              010244                          L10005:
                010244  104423                          TRAP    C$MSG
41
42              010246                          ERR5::
43              010246  004737  011046                  JSR     PC,TYPLOC       ;REPORT THE ADDRESS OF THE FAILURE
44                                                                              ;PRINT 'EXPCTD: XXXXXX RECVED: XXXXXX'
45              010252  013746  002300                  MOV     RECVED,-(SP)
                010256  013746  002276                  MOV     EXPCTD,-(SP)
                010262  012746  004351                  MOV     #FRMT33,-(SP)
                010266  012746  000003                  MOV     #3,-(SP)
                010272  010600                          MOV     SP,R0
                010274  104414                          TRAP    C$PNTB
                010276  062706  000010                  ADD     #10,SP
46              010302  004737  011132                  JSR     PC,DMPREG       ;DUMP THE ERROR REGISTERS
47                                                                              ;CR-LF
48              010306  012746  002656                  MOV     #CRLF,-(SP)
                010312  012746  000001                  MOV     #1,-(SP)
                010316  010600                          MOV     SP,R0
                010320  104414                          TRAP    C$PNTB
                010322  062706  000004                  ADD     #4,SP
49              010326                          L10006:
                010326  104423                          TRAP    C$MSG
50
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;AUTO SIZE FOR RH70 CONTROLLER AND DETERMINE IF IT IS JUMPERED FOR 22 OR
                                ;32 REGISTERS
                                ;CALL
                                ;       JSR     PC,SIZE70       ;CALL ROUTINE
                                ;
                                ;R5 MUST CONTAIN POINTER TO NEW RPCS1 BASE ADDRESS

010330  005037  002430          SIZE70: CLR     RHEXT           ;CLEAR RPBAE OFFSET
010334  005037  002432                  CLR     RHTYPE          ;CLEAR RHXX TYPE REGISTER (RH11)
010340  013746  000004                  MOV     ERRVEC,-(SP)    ;SAVE CONTENTS OF ERROR VECTOR
010344  012737  010414  000004          MOV     #2$,ERRVEC      ;SETUP 'TRAP' RETURN ADDRESS
010352  011500                          MOV     (R5),R0         ;GET RPCS1 ADDRESS
010354  062700  000050                  ADD     #50,R0          ;GET REGISTER OFFSET FOR RH70
010360  012702  000012                  MOV     #10.,R2         ;GET NUMBER OF REGISTERS TO CHECK
010364  005720                          TST     (R0)+           ;TRAP IF NOT A VALID RPBAE
010366  005720                          TST     (R0)+           ;TRAP IF NOT A VALID RPCS3
010370  012737  000050  002430          MOV     #50,RHEXT       ;LOAD OFFSET FOR RPBAE (22 REGISTER RH)
010376  005720                  1$:     TST     (R0)+           ;TRAP IF NOT A VALID REGISTER
010400  005302                          DEC     R2              ;DONE WITH ALL 32 REGISTERS ?
010402  001375                          BNE     1$              ;BR IF NO
010404  012737  000074  002430          MOV     #74,RHEXT       ;LOAD OFFSET FOR RPBAE (32 REGISTER RH)
010412  000403                          BR      3$
010414  012716  010422          2$:     MOV     #3$,(SP)        ;SETUP RETURN ADDRESS
010420  000002                          RTI

010422  011500                  3$:     MOV     (R5),R0         ;GET RPCS1 REGISTER
010424  013702  002430                  MOV     RHEXT,R2        ;GET RPBAE REGISTER OFFSET
010430  001415                          BEQ     4$              ;BR IF NONE
010432  060002                          ADD     R0,R2           ;GET RPBAE REGISTER
010434  052710  001400                  BIS     #A17!A16,(R0)   ;SET EXTENDED ADDRESS BITS IN RPCS1
010440  022712  000003                  CMP     #3,(R2)         ;ARE THE EXTENDED BITS SET IN RPBAE ?
010444  001007                          BNE     4$              ;BR IF NO
010446  005012                          CLR     (R2)            ;CLEAR EXTENDED ADDRESS BITS IN RPBAE
010450  011046                          MOV     (R0),-(SP)      ;SAVE RPCS1 REG CONTENTS
010452  042726  176377                  BIC     #^C<A17!A16>,(SP)+      ;ARE THE EXTEND BITS CLEAR IN RPCS1 ?
010456  001002                          BNE     4$              ;BR IF NO
010460  005237  002432                  INC     RHTYPE          ;SET RHXX TYPE REGISTER (RH70)
010464  012637  000004          4$:     MOV     (SP)+,ERRVEC    ;RESTORE CONTENTS OF ERROR VECTOR
010470  000207                          RTS     PC
```

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

```
                                                BEQ     6$              ;DATA TRANSFER AND BRANCH IF NOT, ELSE
                                                MOV     #DRVCLR,@RPCS1  ;DO A DRIVE CLEAR.
                                                TST     @RPWC           ;IF THE DATA TRANSFER WAS COMPLETED,
                                                BEQ     12$             ;THEN EXIT DRIVER NOW.
                                                CMP     #RDHD,FUNCTN    ;SEE IF COMMAND WAS A READ HEADER, ELSE
                                                BNE     12$             ;EXIT DRIVER IF NOT.
                                                BR      3$              ;CONTINUE COMMAND...

                                        6$:     BIT     #WCE,@RPCS2     ;IS 'WCE' SET ?
                                                BEQ     7$              ;BR IF NO
                                                MOV     #7,ERRMSK       ;LOG 'WRITE CHECK ERROR' STATUS
                                                BR      12$
                                        7$:     MOV     #14,ERRMSK      ;LOG 'TRANSFER ERROR' STATUS
                                                BR      12$

                                        8$:     CMP     FUNCTN,#DIAG    ;WAS IT A "HOUSEKEEP" COMMAND ?
                                                BLOS    10$             ;BR IF NO
                                                TST     @RPDS           ;DID WE GET AN UNEXPECTED ATA ?
                                                BPL     9$              ;BR IF NO
                                                MOV     #6,ERRMSK       ;LOG 'UNEXPECTED ATTENTION ERROR' STATUS
                                                BR      12$

                                        9$:     TST     @RPCS1          ;IS SPECIAL CONDITION SET ?
                                                BPL     10$             ;BR IF NO
                                                MOV     #15,ERRMSK      ;LOG 'SPECIAL CONDITION ERROR' STATUS
                                                BR      12$

                                        10$:    BIT     #ERR,@RPDS      ;DID WE GET A COMPOSITE ERROR?
                                                BEQ     13$             ;BR IF NO
                                        11$:    MOV     #2,ERRMSK       ;LOG 'COMPOSITE ERROR' STATUS
                                        12$:    RTS     PC              ;EXIT

                                        13$:    TST     NOWRCK          ;INHIBIT WRITE CHECK OPERATION ?
                                                BGT     12$             ;BR IF YES
                                                CMP     #FMTRK,FUNCTN   ;WAS THE LAST FUNCTION A FORMAT TRACK ?
                                                BNE     12$             ;BR IF NO
                                                JSR     PC,WRITCK       ;DO THE WRITE-CHECK OPERATION
                                                BR      DRIVER
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE IS USED BY THE ERROR REPORTING SECTION OF THE PROGRAM. THE
 4                                ;PURPOSE OF THIS MODULE IS TO REPORT THE FUNCTION IN PROCESS AT THE TIME OF
 5                                ;FAILURE, THE TRACK AND CYLINDER IN PROCESS AT THE TIME OF THE FAILURE.
 6                                ;*
 7                                ;INPUTS ARE:    ERRFNC, DESTRK+1, DESCYL
 8                                ;
 9                                ;OUTPUT IS USER CONSUMABLE.
10                                ;*
11                                ;CALL
12                                ;       JSR     PC,TYPLOC
13                                ;;*****************************************************************************
14
15 011046                        TYPLOC:                         ;PRINT 'PRSNT FUN.:  '
16 011046  013746  002260                MOV     ERRFNC,-(SP)
   011052  012746  004070                MOV     #FRMT22,-(SP)
   011056  012746  000002                MOV     #2,-(SP)
   011062  010600                        MOV     SP,R0
   011064  104414                        TRAP    C$PNTB
   011066  062706  000006                ADD     #6,SP
17
18 011072  113737  002315  002362        MOVB    DESTRK+1,CONTLT ;GET THE TRACK ADDRESS
19                                                               ;PRINT 'CYL:    TRK: '
20 011100  013746  002362                MOV     CONTLT,-(SP)
   011104  013746  002312                MOV     DESCYL,-(SP)
   011110  012746  004034                MOV     #FRMT20,-(SP)
   011114  012746  000003                MOV     #3, (SP)
   011120  010600                        MOV     SP,R0
   011122  104414                        TRAP    C$PNTB
   011124  062706  000010                ADD     #10,SP
21 011130  000207                        RTS     PC              ;RETURN
```

```
 1
 2                              ;;*****************************************************************************
 3                              ;THIS MODULE IS USED BY THE ERROR REPORTING SECTION OF THE PROGRAM. IT
 4                              ;PRODUCES A REGISTER SNAPSHOT OF THE RHXX REGISTERS SUBSEQUENT TO A DETECTABLE
 5                              ;ERROR.
 6                              ;*
 7                              ;INPUTS ARE:    FROM REG - REG+52
 8                              ;
 9                              ;THE OUTPUT IS USER CONSUMABLE.
10                              ;*
11                              ;CALL
12                              ;       JSR     PC,DMPREG
13                              ;;*****************************************************************************
14
15 011132                      DMPREG:                         ;PRINT 'DRIVE  RPCS1  RPWC   RPBA   RPDA   RPCS2  RPDS'
16 011132  012746  002664              MOV     #FRMT00,-(SP)
   011136  012746  000001              MOV     #1,-(SP)
   011142  010600                      MOV     SP,R0
   011144  104415                      TRAP    C$PNTX
   011146  062706  000004              ADD     #4,SP
17 011152  013746  002526              MOV     REG+12,-(SP)
   011156  013746  002524              MOV     REG+10,-(SP)
   011162  013746  002522              MOV     REG+6,-(SP)
   011166  013746  002520              MOV     REG+4,-(SP)
   011172  013746  002516              MOV     REG+2,-(SP)
   011176  013746  002514              MOV     REG,-(SP)
   011202  013746  002434              MOV     DRVNO,-(SP)
   011206  012746  002755              MOV     #FRMT01,-(SP)
   011212  012746  000010              MOV     #10,-(SP)
   011216  010600                      MOV     SP,R0
   011220  104415                      TRAP    C$PNTX
   011222  062706  000022              ADD     #22,SP
18                                                      ;PRINT 'RPER1  RPAS   RPLA   RPDB   RPMR1  RPDT   RPSN'
19 011226  012746  003035              MOV     #FRMT02,-(SP)
   011232  012746  000001              MOV     #1,-(SP)
   011236  010600                      MOV     SP,R0
   011240  104415                      TRAP    C$PNTX
   011242  062706  000004              ADD     #4,SP
20 011246  013746  002544              MOV     REG+30,-(SP)
   011252  013746  002542              MOV     REG+26,-(SP)
   011256  013746  002540              MOV     REG+24,-(SP)
   011262  013746  002536              MOV     REG+22,-(SP)
   011266  013746  002534              MOV     REG+20,-(SP)
   011272  013746  002532              MOV     REG+16,-(SP)
   011276  013746  002530              MOV     REG+14,-(SP)
   011302  012746  003126              MOV     #FRMT03,-(SP)
   011306  012746  000010              MOV     #10,-(SP)
   011312  010600                      MOV     SP,R0
   011314  104415                      TRAP    C$PNTX
   011316  062706  000022              ADD     #22,SP
21                                                      ;PRINT 'RPOF   RPDC   RPCC   RPER2  RPER3  RPEC1  RPEC2'
22 011322  012746  003206              MOV     #FRMT04,-(SP)
   011326  012746  000001              MOV     #1,-(SP)
   011332  010600                      MOV     SP,R0
   011334  104415                      TRAP    C$PNTX
   011336  062706  000004              ADD     #4,SP
23 011342  013746  002562              MOV     REG+46,-(SP)
```

```
        011346  013746  002560                 MOV     REG+44,-(SP)
        011352  013746  002556                 MOV     REG+42,-(SP)
        011356  013746  002554                 MOV     REG+40,-(SP)
        011362  013746  002552                 MOV     REG+36,-(SP)
        011366  013746  002550                 MOV     REG+34,-(SP)
        011372  013746  002546                 MOV     REG+32,-(SP)
        011376  012746  003300                 MOV     #FRMT05,-(SP)
        011402  012746  000010                 MOV     #10,-(SP)
        011406  010600                         MOV     SP,R0
        011410  104415                         TRAP    C$PNTX
        011412  062706  000022                 ADD     #22,SP
    24
    25  011416  005737  002432                 TST     RHTYPE          ;IS IT RH70 CONTROLLER ?
    26  011422  001424                         BEQ     1$              ;BR IF NO
    27                                                                  ;PRINT 'RPBAE  RPCS3
    28  011424  012746  003362                 MOV     #FRMT06,-(SP)
        011430  012746  000001                 MOV     #1,-(SP)
        011434  010600                         MOV     SP,R0
        011436  104415                         TRAP    C$PNTX
        011440  062706  000004                 ADD     #4,SP
    29  011444  013746  002566                 MOV     REG+52,-(SP)
        011450  013746  002564                 MOV     REG+50,-(SP)
        011454  012746  003402                 MOV     #FRMT07,-(SP)
        011460  012746  000003                 MOV     #3,-(SP)
        011464  010600                         MOV     SP,R0
        011466  104415                         TRAP    C$PNTX
        011470  062706  000010                 ADD     #10,SP
    30  011474  000207              1$:        RTS     PC              ;RETURN
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE IS USED TO LOG DATA INTO THE BAD SECTOR FILE.  THIS DATA IS PUT
 4                                ;INTO THE BSF BECAUSE IT WON'T FIT INTO THE TD FOR AN EXISTING TRACK.
 5                                ;*
 6                                ;INPUTS ARE:    FROM TMPBSF, TMPBSF+2
 7                                ;
 8                                ;OUTPUTS ARE:   TO A BUFFER CALLED 'MFGBUF' (OUTPUT IS CONTROLLED BY 'ENTBSF',
 9                                ;               INDICATING THAT THE BSF IS ALREADY FULL)
10                                ;*
11                                ;CALL
12                                ;       JSR     PC,LODBSF       ;CALL ROUTINE
13                                ;;*****************************************************************************
14
15 011476                        LODBSF:
   011476  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
16 011500  005737  002170                TST     MODTD           ;SEE IF MANUAL MODIFY IS ENABLED,
17 011504  001404                        BEQ     1$              ;BRANCH IF NOT, ELSE
18 011506  005737  002202                TST     MODBY           ;CHECK TO SEE IF MODIFIING BY WORDS OR INDEX,
19 011512  001404                        BEQ     2$              ;BRANCH IF WORD COUNT MODE, ELSE
20 011514  000442                        BR      5$              ;MUST BE INDEX MODE. SO EXIT.
21
22 011516  005737  002200        1$:     TST     ENWTTD          ;SEE IF WRT TD IS ENABLED DURING SCAN MODE,
23 011522  001437                        BEQ     5$              ;BRANCH IF NOT, ELSE
24 011524  013702  011626        2$:     MOV     ENTBSF,R2       ;GET POINTER FOR ENTRY INTO BSF
25 011530  020227  032462                CMP     R2,#ENDBSF      ;IS BSF BUFFER FULL ?
26 011534  101020                        BHI     4$              ;BR IF YES
27 011536  023737  031460  002222        CMP     TMPBSF,LSCYL    ;SEE WHAT CYLINDER ADDRESS IS BEING FLAGGED,
28 011544  101026                        BHI     5$              ;BRANCH IF FE CYLINDER(S),
29 011546  103404                        BLO     3$              ;BRANCH IF ANY USER CYLINDER, ELSE
30 011550  123737  031463  002224        CMPB    TMPBSF+3,LSTRK  ;SEE IF WE ARE FLAGGING THE BSF AREA,
31 011556  001421                        BEQ     5$              ;BRANCH IF SO, ELSE
32 011560  013722  031460        3$:     MOV     TMPBSF,(R2)+    ;LOG THE BSF DATA.
33 011564  013722  031462                MOV     TMPBSF+2,(R2)+  ;CYLINDER/TRACK/SECTOR
34 011570  010237  011626                MOV     R2,ENTBSF       ;UPDATE POINTER FOR NEXT ENTRY INTO BSF
35 011574  000412                        BR      5$              ;AND FOR NOW, TAKE THE RETURN
36 011576                        4$:                             ;PRINT 'WARNING ** BAD SECTOR FILE BUFFER IS
37                                                               ;       FULL, NO MORE ENTRIES ALLOWED'
38 011576  012746  006206                MOV     #WARN,-(SP)
   011602  012746  005241                MOV     #FRMT63,-(SP)
   011606  012746  000002                MOV     #2,-(SP)
   011612  010600                        MOV     SP,R0
   011614  104417                        TRAP    C$PNTF
   011616  062706  000006                ADD     #6,SP
39 011622                        5$:
   011622  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
40 011624  000207                        RTS     PC              ;RETURN TO THE CALLER
41
42 011626  000000        ENTBSF: .WORD   0               ;CONTAINS ADDRESS POINTER FOR NEXT ENTRY INTO
43                                                       ;THE BAD SECTOR FILE.
```

```
 1
 2                              ;;*****************************************************************************
 3                              ;THIS MODULE DOES THE BUFFER CHANGING WHEN A WRITE-CHECK OR READ HEADER & DATA
 4                              ;OPERATION IS REQUIRED ON A DRIVE WHICH WAS FORMATTED IN INTERLEAVED MODE.
 5                              ;THE WRITE CHECK AND READ HEADER & DATA FUNCTION ARE ALWAY DONE IN A LINEAR
 6                              ;FASHION, IE: SECTOR (N), SECTOR (N+1), SECTOR (N+2), SECTOR (N+...),
 7                              ;REGARDLESS OF WHETHER THE DRIVE IS IN INTERLEAVED OR NON-INTERLEAVED MODE.
 8                              ;*
 9                              ;INPUTS ARE:    HDRBLK
10                              ;
11                              ;OUTPUTS ARE:   LINBUF
12                              ;*
13                              ;THIS MODULE IS CALLED BY: WRITCK, VERIFY
14                              ;;*****************************************************************************
15
16 011630                      CHABUF:
   011630  010146                      MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   011632  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   011634  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
   011636  010446                      MOV     R4,-(SP)        ;;PUSH R4 ON STACK
   011640  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
17 011642  012701  032464              MOV     #HDRBLK,R1      ;GET POINTER TO BEGINNING OF INTERLEAVED DATA
18 011646  012702  034744              MOV     #LINBUF,R2      ;GET POINTER TO BEGINNING OF LINEAR BUFFER
19 011652  012703  000002              MOV     #2,R3           ;AND DO FOR 2 ITERATIONS
20 011656  113705  002322      1$:     MOVB    SEC50,R5        ;GET THE # OF SECTORS
21 011662  005205                      INC     R5              ;+1
22 011664  006205                      ASR     R5              ;AND DIVIDE BY 2
23 011666  012704  000006      2$:     MOV     #6,R4           ;# OF WORDS/SECTOR
24 011672  012122              3$:     MOV     (R1)+,(R2)+     ;LOG THE DATA
25 011674  005304                      DEC     R4              ;REDUCE # OF REMAINING WORDS
26 011676  003375                      BGT     3$              ;IF > 0, DO AGAIN
27 011700  062701  000014              ADD     #12.,R1         ;SKIP A SECTOR
28 011704  005305                      DEC     R5              ;ONE LESS SECTOR TO DO
29 011706  003367                      BGT     2$              ;IF > 0, KEEP ON GOING
30 011710  012701  032500              MOV     #HDRBLK+12.,R1  ;GET SECOND HALF OF BUFFER
31 011714  005303                      DEC     R3              ;SECOND PASS, R3=1
32 011716  003357                      BGT     1$              ;IF > 0, IT TIME FOR SECOND PASS
33 011720  012605                      MOV     (SP)+,R5        ;;POP STACK INTO R5
   011722  012604                      MOV     (SP)+,R4        ;;POP STACK INTO R4
   011724  012603                      MOV     (SP)+,R3        ;;POP STACK INTO R3
   011726  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
   011730  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
34 011732  000207                      RTS     PC              ;RETURN
```

```
                                .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:   BUSADR, FUNCTN, ERRFNC, RPOF, NEGWRD
                                ;*
                                ;THIS ROUTINE IS CALLED FROM THE 'DRIVER'.
                                ;;*****************************************************************************
                                ;
011734  105737  002244        WRITCK: TSTB    INTLEV          ;INTERLEAVE MODE ?
011740  001405                        BEQ     1$              ;BR IF NO
011742  004737  011630                JSR     PC,CHABUF       ;RE-SHUFFLE INTERLEAVED DATA AND
011746  012737  034744  002234        MOV     #LINBUF,BUSADR  ;GET POINTER TO BEGINNING OF LINEAR BUFFER
011754  012737  000053  002404  1$:   MOV     #WCKHD,FUNCTN   ;SETUP A WRITE-CHECK HEADERS OPERATION
011762  012737  005653  002260        MOV     #WCKHDR,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
011770  052737  100000  002236        BIS     #CMOD,OFFSET    ;SET COMMAND MODIFIER BIT
011776  013737  002364  002376        MOV     TKWCNT,NEGWRD   ;AND SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
012004  000207                        RTS     PC              ;NOW TAKE THE RETURN
```

```
                                .SBTTL  MANUAL INPUT ROUTINE
                                ;;*****************************************************************************
                                ;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 TRACK
                                ;DESCRIPTOR.
                                ;*
                                ;INPUTS ARE:    TDCPY3 - TDCPY6, DESCYL, DESTRK, @NULL
                                ;
                                ;OUTPUTS ARE:   TDCPY3 - TDCPY6, TDWRD1 - TDWRD6, TEXT, RELWRD, SECADD
                                ;*
                                ;THIS MODULE IS CALLED BY: MAIN
                                ;;*****************************************************************************

012006                          INPUTD:
012006  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012010  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
                                                               ;PRINT 'MODIFY BY (0=WORDS, 1=INDEX, 2=CHANGE)?'
012012  104443                        TRAP    C$GMAN
012014  000406                        BR      10000$
012016  002202                        .WORD   MODBY
012020  000052                        .WORD   T$CODE
012022  006570                        .WORD   MESG22
012024  000003                        .WORD   3
012026  000000                        .WORD   T$LOLIM
012030  000002                        .WORD   T$HILIM
012032                          10000$:
012032  005737  002156                TST     ANYWHR          ;DON'T ALLOW DESIRED CYLINDER TO BE CHANGED,
012036  001410                        BEQ     1$              ;IF WRITTING ON FE CYLINDER ONLY, ELSE
                                                               ;PRINT 'CYLINDER ADDRESS (D)  0 ?'
012040  104443                        TRAP    C$GMAN
012042  000406                        BR      10001$
012044  002312                        .WORD   DESCYL
012046  000052                        .WORD   T$CODE
012050  012654                        .WORD   MESG38
012052  177777                        .WORD   177777
012054  000000                        .WORD   T$LOLIM
012056  001166                        .WORD   T$HILIM
012060                          10001$:
012060                          1$:                             ;PRINT 'TRACK ADDRESS (D)  0 ?'
012060  104443                        TRAP    C$GMAN
012062  000406                        BR      10002$
012064  002314                        .WORD   DESTRK
012066  000052                        .WORD   T$CODE
012070  012675                        .WORD   MESG39
012072  177400                        .WORD   177400
012074  000000                        .WORD   T$LOLIM
012076  000037                        .WORD   T$HILIM
012100                          10002$:

012100  004737  021440                JSR     PC,READTD       ;CALL THE READ TD SUBROUTINE
012104  000401                        BR      2$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
012106  000402                        BR      3$              ;ELSE RETURN HERE, NO ERROR
012110  004737  026332          2$:   JSR     PC,DESTD        ;GET DESIRED TD, BECAUSE WE FAILED THE READ

012114  012737  006030  002360  3$:   MOV     #CURENT,TEXT    ;LOAD TEXT WITH ADDRESS OF MESSAGE
012122  004737  017240                JSR     PC,TDDUMP       ;GO DUMP THE TRACK DESCRIPTOR
```

MANUAL INPUT ROUTINE

```
                                                CMP     MODBY,#1        ;MODIFY BY WORDS, INDEX OR CHANGE MODE ?
                                                BLT     6$              ;BR IF WORDS
                                                BEQ     7$              ;BR IF INDEX
                                                                        ;MUST BE CHANGE MODE
                                                BIC     #BIT14!BIT15,TDWRD1     ;RESET TD & HDR 0 MOVED BITS
                                                CLR     TDCPY3          ;PUT DUMMY NULL IN TD WORD #3
                                                CLR     TDCPY4          ;PUT DUMMY NULL IN TD WORD #4
                                                CLR     TDCPY5          ;PUT DUMMY NULL IN TD WORD #5
                                                CLR     TDCPY6          ;PUT DUMMY NULL IN TD WORD #6
                                                                        ;PRINT 'TYPE <CR> TO INPUT (0-140000) DATA;'
                                                MOV     #FRMT24,-(SP)
                                                MOV     #1,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTB
                                                ADD     #4,SP
                                                                        ;PRINT 'TD WORD#3 (D)  0 ?'
                                                TRAP    C$GMAN
                                                BR      10003$
                                                .WORD   TDCPY3
                                                .WORD   T$CODE
                                                .WORD   MESG43
                                                .WORD   177777
                                                .WORD   T$LOLIM
                                                .WORD   T$HILIM
                                        10003$:
                                                TST     TDCPY3          ;NULL ENTRY ?
                                                BEQ     5$              ;BR IF YES
                                                                        ;PRINT 'TD WORD#4 (D)  0 ?'
                                                TRAP    C$GMAN
                                                BR      10004$
                                                .WORD   TDCPY4
                                                .WORD   T$CODE
                                                .WORD   MESG44
                                                .WORD   177777
                                                .WORD   T$LOLIM
                                                .WORD   T$HILIM
                                        10004$:
                                                TST     TDCPY4          ;NULL ENTRY ?
                                                BEQ     5$              ;BR IF YES
                                                                        ;PRINT 'TD WORD#5 (D)  0 ?'
                                                TRAP    C$GMAN
                                                BR      10005$
                                                .WORD   TDCPY5
                                                .WORD   T$CODE
                                                .WORD   MESG45
                                                .WORD   177777
                                                .WORD   T$LOLIM
                                                .WORD   T$HILIM
                                        10005$:
                                                TST     TDCPY5          ;NULL ENTRY ?
                                                BEQ     5$              ;BR IF YES
                                                                        ;PRINT 'TD WORD#6 (D)  0 ?'
                                                TRAP    C$GMAN
                                                BR      10006$
                                                .WORD   TDCPY6
                                                .WORD   T$CODE
                                                .WORD   MESG46
```

```
        012322  177777                          .WORD   177777
        012324  000001                          .WORD   T$LOLIM
        012326  042077                          .WORD   T$HILIM
        012330                          10006$:
     65 012330  013737  031434  031420  5$:     MOV     TDCPY3,TDWRD3   ;GET TD WORD #3
     66 012336  013737  031436  031422          MOV     TDCPY4,TDWRD4   ;GET TD WORD #4
     67 012344  013737  031440  031424          MOV     TDCPY5,TDWRD5   ;GET TD WORD #5
     68 012352  013737  031442  031426          MOV     TDCPY6,TDWRD6   ;GET TD WORD #6
     69 012360  000454                          BR      9$
     70 012362                          6$:                             ;PRINT 'SECTOR ADDRESS (D) ?'
     71 012362  104443                          TRAP    C$GMAN
        012364  000406                          BR      10007$
        012366  002402                          .WORD   SECADD
        012370  000042                          .WORD   T$CODE
        012372  012753                          .WORD   MESG42
        012374  177777                          .WORD   177777
        012376  000000                          .WORD   T$LOLIM
        012400  000061                          .WORD   T$HILIM
        012402                          10007$:
     72                                                                 ;PRINT 'WORD IN SECTOR (D) ?'
     73 012402  104443                          TRAP    C$GMAN
        012404  000406                          BR      10010$
        012406  002400                          .WORD   RELWRD
        012410  000042                          .WORD   T$CODE
        012412  012713                          .WORD   MESG40
        012414  177777                          .WORD   177777
        012416  000000                          .WORD   T$LOLIM
        012420  000401                          .WORD   T$HILIM
        012422                          10010$:
     74 012422  004737  016536                  JSR     PC,RELTIVE      ;CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
     75 012426  023737  002400  002354          CMP     RELWRD,TDLMTS+2 ;WILL BEGINNING OF DEFECT BE IN GAP 2 ?
     76 012434  003016                          BGT     8$              ;BR IF NO
     77 012436  013737  002354  002400          MOV     TDLMTS+2,RELWRD ;GET UPPER LIMIT FOR HDR 0 MOVED AND
     78 012444  005237  002400                  INC     RELWRD          ;ADD ONE TO IT.
     79 012450  000410                          BR      8$              ;AND GO INSERT IT INTO TD.
     80 012452                          7$:                             ;PRINT 'WORDS FROM INDEX (D) ?'
     81 012452  104443                          TRAP    C$GMAN
        012454  000406                          BR      10011$
        012456  002400                          .WORD   RELWRD
        012460  000042                          .WORD   T$CODE
        012462  012732                          .WORD   MESG41
        012464  177777                          .WORD   177777
        012466  000166                          .WORD   T$LOLIM
        012470  042553                          .WORD   T$HILIM
        012472                          10011$:
     82 012472  004737  016616          8$:     JSR     PC,INSERT       ;INSERT THE DEFECT IN TD
     83 012476  005737  031460                  TST     TMPBSF          ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
     84 012502  100403                          BMI     9$              ;BR IF YES
     85 012504  004737  011476                  JSR     PC,LODBSF       ;LOAD THE BSF BUFFER.
     86 012510  000446                          BR      16$
     87
     88 012512  012700  031420          9$:     MOV     #TDWRD3,R0      ;GET POINTER TO BEGINNING OF TD BUFFER
     89 012516  012701  000004                  MOV     #4,R1           ;# OF DEFECT WORDS IN TD
     90 012522  005720                  10$:    TST     (R0)+           ;IS THIS A DUMMY (0) ENTRY ?
     91 012524  001003                          BNE     11$             ;BR IF NO
     92 012526  012760  140000  177776          MOV     #NULL,-2(R0)    ;MAKE DUMMY (0) ENTRY A NULL (140000) ENTRY
     93 012534  005301                  11$:    DEC     R1              ;DONE ALL WORDS YET ?
```

```
                                        BGT     10$             ;BR IF NO
                                        MOV     ERRMAX,R0       ;SETUP RETRY COUNT
                                        JSR     PC,SAVETD       ;SAVE TD WRITE BUFFER JUST INCASE THERE
                                                                ;IS RETRY ATTEMPT.
                                12$:    JSR     PC,RESTTD       ;NOW RESTORE THE TD BUFFER
                                        JSR     PC,WRITTD       ;CALL THE WRITE TD SUBROUTINE
                                        BR      15$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE, NO ERROR
                                        MOV     #-1,SUPRSS      ;SUPRESS THE ERROR MESSAGE OUTPUT
                                13$:    DEC     R0              ;DID WE EXCEED RETRY ?
                                        BLT     15$             ;BR IF YES
                                        BGT     14$             ;BR IF NO
                                        CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
                                14$:    JSR     PC,READTD       ;CALL THE READ TD SUBROUTINE
                                        BR      13$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE, NO ERROR
                                        MOV     #CHANGE,TEXT    ;LOAD TEXT WITH ADDRESS OF MESSAGE
                                        JSR     PC,TDDUMP       ;GO DUMP THE TRACK DESCRIPTOR
                                        BR      16$
                                15$:    TST     (R5)+           ;ADJUST R5 TO SKIP FORMAT TRACK, ON RETURN
                                16$:    CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                                                ;PRINT 'DO YOU WANT TO MODIFY ANYMORE TD'S (L)  N ?'
                                        TRAP    C$GMAN
                                        BR      10012$
                                        .WORD   MORETD
                                        .WORD   T$CODE
                                        .WORD   MESG47
                                        .WORD   1
                                10012$:
                                        RTS     PC              ;TAKE THE RETURN

012654  103     131     114     MESG38: .ASCIZ  /CYLINDER ADDRESS/
012675  124     122     101     MESG39: .ASCIZ  /TRACK ADDRESS/
012713  127     117     122     MESG40: .ASCIZ  /WORD IN SECTOR/
012732  127     117     122     MESG41: .ASCIZ  /WORDS FROM INDEX/
012753  123     105     103     MESG42: .ASCIZ  /SECTOR ADDRESS/
012772  124     104     040     MESG43: .ASCIZ  /TD WORD#3/
013004  124     104     040     MESG44: .ASCIZ  /TD WORD#4/
013016  124     104     040     MESG45: .ASCIZ  /TD WORD#5/
013030  124     104     040     MESG46: .ASCIZ  /TD WORD#6/
013042  104     117     040     MESG47: .ASCIZ  /DO YOU WANT TO MODIFY ANYMORE TD'S/

                                        .EVEN
```

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

013106                          WAIT:
013106  012727  000372                  MOV     #250.,(PC)+
013112  000000                         .WORD   0
013114  013727  002116                  MOV     L$DLY,(PC)+
013120  000000                         .WORD   0
013122  005367  177772                  DEC      6(PC)
013126  001375                         BNE     .-4
013130  005367  177756                  DEC     -22(PC)
013134  001367                         BNE     .-20
013136  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
                                ;;*****************************************************************************

013140  052777  000040  167302  SEIZE:  BIS     #CLR,@RPCS2     ;CLEAR THE MASSBUS
013146  013777  002434  167274          MOV     DRVNO,@RPCS2    ;RELOAD THE DRIVE NUMBER
013154  105777  167272                  TSTB    @RPDS           ;IS THIS DRIVE SEIZED BY ANOTHER PORT?
013160  100435                          BMI     3$              ;NO, JUST TAKE RETURN
013162  005077  167264                  CLR     @RPDS           ;ISSUE A DRIVE REQUEST
013166  010246                          MOV     R2,-(SP)        ;SAVE R2
013170  012702  000012                  MOV     #10.,R2         ;LOAD R2 WITH AN OVERAL ITERATION COUNT
013174  005777  167252          1$:     TST     @RPDS           ;NOW WAIT FOR THE OTHER PORT TO RELEASE
013200  001021                          BNE     2$              ;NOT YET IF RPDS = 0
013202  004737  013106                  JSR     PC,WAIT
013206  005302                          DEC     R2              ;AND REDUCE THE ITERATION COUNT
013210  001371                          BNE     1$              ;LOOK AGAIN FOR DRIVE PRESENT
013212  012737  000003  002272          MOV     #3,ERRMSK       ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
013220  013746  002252                  MOV     SUPRSS,-(SP)    ;;PUSH SUPRSS ON STACK
013224  005037  002252                  CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
013230  004737  017542                  JSR     PC,SAVRPR       ;SAVE RP07 REGISTERS ON ERROR
013234  004737  013402                  JSR     PC,ERRORS       ;AND REPORT THE ERROR
013240  012637  002252                  MOV     (SP)+,SUPRSS    ;;POP STACK INTO SUPRSS

013244  112777  000377  167204  2$:     MOVB    #377,@RPAS      ;CLEAR ANY UNWANTED ATTENTION BITS
013252  012602                          MOV     (SP)+,R2        ;RESTORE R2
013254  000207                  3$:     RTS     PC              ;AND TAKE RETURN
```

```
                                ;;*****************************************************************************
                                ;THIS MODULE, WHEN CALLED DOES A CLEAR TO THE DRIVE IT IS DEPENDENT ON THE
                                ;MODULE "SEIZE  TO RELOAD THE DRIVE NUMBER AFTER THE RESET.
                                ;*
                                ;INPUTS ARE:    NONE
                                ;
                                ;OUTPUTS ARE:   RPCS1, RPOF, RPMR1
                                ;*
                                ;THIS MODULE IS CALLED BY: RETRY
                                ;;*****************************************************************************

013256  004737  013140          RESET:  JSR     PC,SEIZE        ;NOW GET THE DRIVE
013262  012777  000021  167150          MOV     @READIN,@RPCS1  ;DO A READ IN PRESET
013270  012737  010000  002236          MOV     @FMT16,OFFSET   ;SET UP FOR 16 BIT WORD
013276  004737  010472                  JSR     PC,READY        ;WAIT FOR DRIVE READY NOW!
013302  000207                          RTS     PC              ;NOW RETURN TO MAIN
```

```
                                .SBTTL  RETRY ROUTINE
                                ;
                                ;;*****************************************************************************
                                ;THIS ROUTINE MAINTAINS THE RETRY COUNTER.  RETRY MAXIMUM IS 'ERRMAX'.  AN
                                ;ERROR WILL BE REPORTED IF THE MAXIMUM RETRY COUNT IS EXCEEDED.
                                ;*
                                ;INPUTS ARE:    ERRMSK
                                ;
                                ;OUTPUTS ARE:   ERRMSK, RTYCNT, ERRTTL
                                ;*
                                ;CALL
                                ;       JSR     PC,RETRY        ;CALL THE ROUTINE
                                ;       ------                  ;RETURN HERE IF EXCEEDED MAX RETRY LIMIT
                                ;       - ----                  ;ELSE RETURN HERE, IF RTYCNT >= 1 THEN ERROR
                                ;                                               IF RTYCNT  = 0 THEN NO ERROR
                                ;*
                                ;THIS MODULE IS MAINLY CALLED BY: MAKTRK, DRVINI, RDBSF, WRTBSF, WRITTD,
                                ;                                 READTD, HDSCAN, DASCAN
                                ;;*****************************************************************************

013304  005337  002264        RETRY:  DEC     ITCNT           ;DECREMENT ITERATION FOR THIS COMMAND
013310  062716  000002                ADD     #2,(SP)         ;ADJUST FOR GOOD RETURN
013314  004737  017542                JSR     PC,SAVRPR       ;SAVE RP07 REGISTERS
013320  005737  002272                TST     ERRMSK          ;DID WE GET ANY ERRORS ?
013324  001414                        BEQ     1$              ;BR IF NO
013326  005237  002274                INC     RTYCNT          ;INCREMENT THE LOCAL RETRY COUNTER AND
013332  005237  002270                INC     ERRTTL          ;THE TOTAL ERROR COUNTER.
013336  023737  002274  002226        CMP     RTYCNT,ERRMAX   ;DID WE EXCEED THE MAX RETRY LIMIT ON ERROR ?
013344  103411                        BLO     2$              ;BR IF NO
013346  162716  000002                SUB     #2,(SP)         ;ADJUST FOR EXCEEDED RETRY LIMIT RETURN
013352  004737  013402                JSR     PC,ERRORS       ;REPORT THE ERROR NOW!

013356  005037  002274        1$:     CLR     RTYCNT          ;RESET RETRY COUNT TO 0
013362  005737  002272                TST     ERRMSK          ;DID WE GET AN ERROR ?
013366  001404                        BEQ     3$              ;BR IF NO
013370  005037  002272        2$:     CLR     ERRMSK          ;RESET ERROR STATUS AND
013374  004737  013256                JSR     PC,RESET        ;RESET THE MASSBUS.
013400  000207                3$:     RTS     PC              ;EXIT
```

```
                                .SBTTL  ERROR HANDLER
                                ;
                                ;;*****************************************************************************
                                ;THIS ROUTINE LOOKS AT ERRMSK, AND DECODES THE BIT MASK BITS 0-15 ARE
                                ;APPLICABLE.
                                ;*
                                ;INPUTS ARE:    ERRMSK
                                ;
                                ;OUTPUTS ARE:   ERRMSK, RTYCNT, ERRTTL
                                ;*
                                ;THIS MODULE IS CALLED BY: RETRY
                                ;;*****************************************************************************
                                ;
013402  005737  002272  ERRORS: TST     ERRMSK          ;DID WE GET ANY ERRORS?
013406  001430                  BEQ     4$              ;IF ERRMSK = 0, NO!
013410  005737  002252          TST     SUPRSS          ;ALLOW ERROR MESSAGES ?
013414  100425                  BMI     4$              ;IF MINUS, NO....
013416  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013420  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013422  012701  013472          MOV     #5$,R1          ;GET THE TOP OF THE ERROR DISPATCH FILE
013426  012702  000001          MOV     #1,R2           ;AND SET UP THE ERROR NUMBER MASK.
013432  023702  002272  1$:     CMP     ERRMSK,R2       ;MATCH?
013436  001407                  BEQ     2$              ;TAKE BRANCH OF DS - REPORT FIND NOW!
013440  005202                  INC     R2              ;GET NEXT ERROR POSITION
013442  062701  000012          ADD     #12,R1          ;MOVE THE DISPATCH POINTER TO NEXT MESSAGE
013446  020127  013732          CMP     R1,#6$          ;ARE WE AT THE END OF ERROR TABLE ?
013452  103767                  BLO     1$              ;BR IF NO
013454  000403                  BR      3$
                                ;
013456  005237  002266  2$:     INC     ERTTL           ;ADD ONE TO THE ERROR TOTAL AND
013462  000111                  JMP     (R1)            ;REPORT THE FINF NOW.
                                ;
013464                  3$:
013464  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
013466  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
013470  000207          4$:     RTS     PC              ;RETURN
                                ;
013472                  5$:     ERRHRD  16,EM16,ERR3    ;TD NOT READABLE THRU
013472  104456                  TRAP    C$ERHRD
013474  000020                  .WORD   16
013476  007454                  .WORD   EM16
013500  010220                  .WORD   ERR3
                                ;                               ;RELOCATION ERROR       ;ERRMSK=1
013502  000770                  BR      3$              ;RETURN
                                ;
013504                          ERRHRD  3,EM3,ERR0      ;COMPOSITE ERROR        ;ERRMSK=2
013504  104456                  TRAP    C$ERHRD
013506  000003                  .WORD   3
013510  007037                  .WORD   EM3
013512  007722                  .WORD   ERR0
013514  000763                  BR      3$              ;RETURN
                                ;
013516                          ERRHRD  4,EM4,ERR0      ;DRIVE HUNG             ;ERRMSK=3
013516  104456                  TRAP    C$ERHRD
013520  000004                  .WORD   4
013522  007063                  .WORD   EM4
013524  007722                  .WORD   ERR0
```

ERROR HANDLER

```
     46 013526  000756                    BR      3$              ;RETURN
     47
     48 013530                            ERRHRD  5,EM5,ERR0      ;DRIVE WRITE LOCKED     ;ERRMSK=4
        013530  104456                    TRAP    C$ERHRD
        013532  000005                    .WORD   5
        013534  007123                    .WORD   EM5
        013536  007722                    .WORD   ERR0
     49 013540  000751                    BR      3$              ;RETURN
     50
     51 013542                            ERRHRD  6,EM6,ERR0      ;DRIVE OFFLINE          ;ERRMSK=5
        013542  104456                    TRAP    C$ERHRD
        013544  000006                    .WORD   6
        013546  007146                    .WORD   EM6
        013550  007722                    .WORD   ERR0
     52 013552  000744                    BR      3$              ;RETURN
     53
     54 013554                            ERRHRD  7,EM7,ERR0      ;UNEXPECTED ATTN        ;ERRMSK=6
        013554  104456                    TRAP    C$ERHRD
        013556  000007                    .WORD   7
        013560  007164                    .WORD   EM7
        013562  007722                    .WORD   ERR0
     55 013564  000737                    BR      3$              ;RETURN
     56
     57 013566                            ERRHRD  10,EM10,ERR0    ;WRITE CHECK ERROR      ;ERRMSK=7
        013566  104456                    TRAP    C$ERHRD
        013570  000012                    .WORD   10
        013572  007215                    .WORD   EM10
        013574  007722                    .WORD   ERR0
     58 013576  000732                    BR      3$              ;RETURN
     59
     60 013600                            ERRHRD  11,EM11,ERR5    ;FORMAT VERIFY ERROR    ;ERRMSK=10
        013600  104456                    TRAP    C$ERHRD
        013602  000013                    .WORD   11
        013604  007243                    .WORD   EM11
        013606  010246                    .WORD   ERR5
     61 013610  000725                    BR      3$              ;RETURN
     62
     63 013612                            ERRHRD  1,EM1,ERR1      ;TD ADDRESS INCORRECT   ;ERRMSK=11
        013612  104456                    TRAP    C$ERHRD
        013614  000001                    .WORD   1
        013616  006766                    .WORD   EM1
        013620  007754                    .WORD   ERR1
     64 013622  000720                    BR      3$              ;RETURN
     65
     66 013624                            ERRHRD  2,EM2,ERR2      ;TD FORMAT INCORRECT    ;ERRMSK=12
        013624  104456                    TRAP    C$ERHRD
        013626  000002                    .WORD   2
        013630  007013                    .WORD   EM2
        013632  010076                    .WORD   ERR2
     67 013634  000713                    BR      3$              ;RETURN
     68
     69 013636                            ERRHRD  12,EM12,ERR0    ;DRY DIDN'T RESET       ;ERRMSK=13
        013636  104456                    TRAP    C$ERHRD
        013640  000014                    .WORD   12
        013642  007267                    .WORD   EM12
        013644  007722                    .WORD   ERR0
     70 013646  000706                    BR      3$              ;RETURN
```

```
71
72 013650                                ERRHRD  13,EM13,ERR0    ;TRANSFER ERROR          ;ERRMSK=14
   013650  104456                        TRAP    C$ERHRD
   013652  000015                        .WORD   13
   013654  007326                        .WORD   EM13
   013656  007722                        .WORD   ERR0
73 013660  000701                        BR      3$              ;RETURN
74
75 013662                                ERRHRD  14,EM14,ERR0    ;SPECIAL CONDITION       ;ERRMSK=15
   013662  104456                        TRAP    C$ERHRD
   013664  000016                        .WORD   14
   013666  007351                        .WORD   EM14
   013670  007722                        .WORD   ERR0
76 013672  000674                        BR      3$              ;RETURN
77
78 013674                                ERRHRD  15,EM15,ERR0    ;RANDOM HEADER ERRORS    ;ERRMSK=16
   013674  104456                        TRAP    C$ERHRD
   013676  000017                        .WORD   15
   013700  007377                        .WORD   EM15
   013702  007722                        .WORD   ERR0
79 013704  000667                        BR      3$              ;RETURN
80
81 013706                                ERRHRD  17,EM17,ERR0    ;UNRECOVERABLE ERROR     ;ERRMSK=17
   013706  104456                        TRAP    C$ERHRD
   013710  000021                        .WORD   17
   013712  007531                        .WORD   EM17
   013714  007722                        .WORD   ERR0
82 013716  000662                        BR      3$              ;RETURN
83
84 013720                                ERRHRD  20,EM20,ERR0    ;RANDOM WRT CHK ERRORS   ;ERRMSK=20
   013720  104456                        TRAP    C$ERHRD
   013722  000024                        .WORD   20
   013724  007576                        .WORD   EM20
   013726  007722                        .WORD   ERR0
85 013730  000655                        BR      3$              ;RETURN
89
90 013732                           6$:
```

```
                                .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 'TDCPY3' (WHICH IS POINTED TO BY R4)
                                ;
                                ;OUTPUTS ARE:   DEFSEC, BUFFER CALLED 'TDCPY3' (WHICH IS POINTED TO BY R4)
                                ;*
                                ;CALL
                                ;       JSR     PC,TDCASE
                                ;;*****************************************************************************

013732  012700  002352          TDCASE: MOV     #TDLMTS,R0      ;GET POINTER TO TD DEFECT LIMITS
013736  012702  000055                  MOV     #<TD+G2>,R2     ;INCLUDE THIS IN THE CALCULATION
013742  021420                          CMP     (R4),(R0)+      ;WAS TD MOVED ?
013744  003426                          BLE     2$              ;BR IF YES
013746  032737  040000  031414          BIT     #BIT14,TDWRD1   ;TD BIT ALREADY SET ?
013754  001022                          BNE     2$              ;BR IF YES
013756  021420                          CMP     (R4),(R0)+      ;WAS HDR 0 MOVED ?
013760  003407                          BLE     1$              ;BR IF YES
013762  032737  100000  031414          BIT     #BIT15,TDWRD1   ;HDR 0 BIT ALREADY SET ?
013770  001003                          BNE     1$              ;BR IF YES
013772  012702  000155                  MOV     #<G1+TD+G2>,R2  ;INCLUDE THIS IN THE CALCULATION
013776  000445                          BR      5$
                                                                ;ACCOUNT FOR HEADER 0 MOVED
014000  005337  002410          1$:     DEC     DEFTRK          ;ONE LESS DEFECT TO GO ON THIS TRACK
014004  162702  000055                  SUB     #<TD+G2>,R2     ;INCLUDE THIS IN THE CALCULATION
014010  016046  177776                  MOV     -2(R0),-(SP)    ;GET HDR 0 MOVED LIMIT
014014  162416                          SUB     (R4)+,(SP)      ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
014016  062602                          ADD     (SP)+,R2        ;AND ADD DIFFERENCE TO THIS CALCULATION.
014020  000420                          BR      3$
                                                                ;ACCOUNT FOR TRACK DESCRIPTOR MOVED
014022  005337  002410          2$:     DEC     DEFTRK          ;ONE LESS DEFECT TO GO ON THIS TRACK
014026  016446  000002                  MOV     2(R4),-(SP)     ;SAVE NEXT DEFECT VALUE
014032  016046  177776                  MOV     -2(R0),-(SP)    ;GET TD MOVED LIMIT
014036  162416                          SUB     (R4)+,(SP)      ;FIND DIFFERENCE, UPDATE POINTER TO NEXT DEFECT
014040  061602                          ADD     (SP),R2         ;AND ADD DIFFERENCE TO THIS CALCULATION.
014042  162616                          SUB     (SP)+,(SP)      ;ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
014044  005720                          TST     (R0)+           ;UPDATE LIMIT POINTER TO LOOK FOR HDR 0 MOVED
014046  022620                          CMP     (SP)+,(R0)+     ;WAS HDR 0 ALSO MOVED ?
014050  003753                          BLE     1$              ;BR IF YES
014052  032737  100000  031414          BIT     #BIT15,TDWRD1   ;HDR 0 BIT ALREADY SET ?
014060  001347                          BNE     1$              ;BR IF YES

014062  021427  140000          3$:     CMP     (R4),#NULL      ;ANY MORE DEFECTS ?
014066  001412                          BEQ     6$              ;BR IF NO
014070  032737  140000  031414          BIT     #BIT14!BIT15,TDWRD1     ;DID TD AND/OR HRD 0 REALLY GET MOVED ?
014076  001003                          BNE     4$              ;BR IF YES
014100  162702  000066                  SUB     #<DS/2>,R2      ;EXCLUDE HALF A SKIP IN THE CALCULATION
014104  000402                          BR      5$
014106  062702  000066          4$:     ADD     #<DS/2>,R2      ;INCLUDE HALF A SKIP IN THE CALCULATION
014112  160214                  5$:     SUB     R2,(R4)         ;SUBTRACT ALONG WITH SKIP DEFECT INFO
014114  000207                  6$:     RTS     PC
```

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

ADJUST DEFECT TO TRACK DESCRIPTOR

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

```
                                .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:    TDWRD1 - TDWRD6, DS, PARTBL, SIZE OF SECTOR (HDR+G3+DATA+G4)
                                ;
                                ;OUTPUTS ARE:   TDSBLK, R0 - R5, DEFSEC, DEFTRK, TDCPY3 - TDCPY6,
                                ;               DEF1 - DEF4, DEFCNT
                                ;*
                                ;THIS MODULE IS CALLED BY: MAKTRK
                                ;;*****************************************************************************

014326  005037  002410        CALCTD: CLR     DEFTRK          ;RESET THE # OF DEFECTS PER TRACK COUNTER
014332  012705  033614                MOV     #TDSBLK,R5      ;POINT TO THE TD OUTPUT BLOCK
014336  013737  002230  002232        MOV     ENDPTR,ENDTAB   ;GET THE BUFFER SIZE LIMIT
014344  006337  002232                ASL     ENDTAB          ;PUT THE LIMIT ON A WORD BOUNDARY
014350  062737  033614  002232        ADD     #TDSBLK,ENDTAB  ;AND SET THE LIMIT
014356  012704  031430        1$:     MOV     #TDCPY1,R4      ;RESTORE POINTER TO 1ST TD WORD
014362  013724  031414                MOV     TDWRD1,(R4)+    ;SAVE TD WORD #1
014366  013724  031416                MOV     TDWRD2,(R4)+    ;SAVE TD WORD #2
014372  013724  031420                MOV     TDWRD3,(R4)+    ;SAVE TD WORD #3
014376  013724  031422                MOV     TDWRD4,(R4)+    ;SAVE TD WORD #4
014402  013724  031424                MOV     TDWRD5,(R4)+    ;SAVE TD WORD #5
014406  013724  031426                MOV     TDWRD6,(R4)+    ;SAVE TD WORD #6
014412  012704  031434                MOV     #TDCPY3,R4      ;RESTORE POINTER TO 3RD TD WORD (1ST DEFECT)
014416  022427  140000                CMP     (R4)+,#NULL     ;ANY DEFECTS ON THIS TRACK ?
014422  001007                        BNE     3$              ;BR IF YES
014424                          2$:                             ;NO MORE DEFECTS ON THIS TRACK, SO NULL THE
                                                                ;SKIP DISPLACEMENT WORDS IN EACH OF THE
014424  012725  140000                MOV     #NULL,(R5)+     ;REMAINING SECTORS.
014430  020537  002232                CMP     R5,ENDTAB       ;DONE WITH THE TABLE YET ?
014434  003773                        BLE     2$              ;BR IF NO
014436  000137  015166                JMP     17$             ;EXIT, GO WRITE FORMAT FOR THIS TRACK
                                                                ;NOW CORRECT EACH SKIP TO THE BEGINNING OF
                                                                ;THE DEFECT, SO IT CAN BE INSERTED INTO IT'S
                                                                ;APPROPRIATE SECTOR.
014442  005237  002410        3$:     INC     DEFTRK          ;ACCUMULATE # OF DEFECTS ON THIS TRACK
014446  023727  002410  000004        CMP     DEFTRK,#4       ;DID WE REACH LAST DEFECT ?
014454  103003                        BHIS    4$              ;BR IF YES
014456  022427  140000                CMP     (R4)+,#NULL     ;ANY MORE DEFECTS ?
014462  001367                        BNE     3$              ;BR IF YES
                                                                ;NOW GO LOOK TO SEE IF THE TD AND/OR HDR 0 HAVE
                                                                ;BEEN MOVED, SO THE REMAINING DEFECTS CAN BE
                                                                ;ADJUSTED ACCORDINGLY.
014464  012704  031434        4$:     MOV     #TDCPY3,R4      ;RESTORE POINTER TO 3RD TD WORD (1ST DEFECT)
014470  004737  013732                JSR     PC,TDCASE       ;LOOK FOR A TD AND/OR HDR 0 MOVED
014474  021427  140000                CMP     (R4),#NULL      ;ANY MORE DEFECTS ?
014500  001751                        BEQ     2$              ;BR IF NO
                                                                ;NOW FIND WHICH SECTOR THE DEFECT SHOULD BE
                                                                ;INSERTED INTO.
014502  162714  000066                SUB     #<DS/2>,(R4)    ;CORRECT 1ST DEFECT TO THE BEGINNING OF DEFECT
014506  105037  002245        5$:     CLRB    SAMSEC          ;RESET THE SAME SECTOR FLAG,
014512  005037  002406                CLR     DEFSEC          ;THE # OF DEFECTS PER SECTOR COUNTER AND
014516  005037  015170                CLR     TMDLTA          ;THE TEMP. ACCUMULATIVE DELTA COUNTER.
```

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

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

```
    172 015170  000000                TMDLTA: .WORD   0           ;TEMP. ACCUMULATIVE DELTA VALUE GOES HERE
```

```
                                .SBTTL  CHECK BAD SECTOR FILE ENTRIES
                                ;
                                ;;**********************************************************************************
                                ;THIS MODULE CHECKS TO SEE IF THE CYLINDER ADDRESS BEING SETUP BY THE 'MAKTRK'
                                ;ROUTINE IS ENTERED IN MFG BAD SECTOR FILE.
                                ;*
                                ;INPUTS ARE:     TMPBSF, TMPBSF+2
                                ;
                                ;OUTPUTS ARE:
                                ;*
                                ;CALL
                                ;       JSR     PC,CHKBSF       ;NOW CHECK THE BAD SECTOR FILE
                                ;       ------                  ;RETURN HERE IF SECTOR NOT FOUND,
                                ;       ------                  ;ELSE RETURN HERE
                                ;*
                                ;THIS MODULE IS CALLED BY: MERGE
                                ;;**********************************************************************************

 015172                         CHKBSF:
 015172  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 015174  012700  031472                 MOV     #MFGBUF+6.,R0   ;GET OVER SERIAL NUMBER IN BSF TABLE
 015200  005720                 1$:     TST     (R0)+           ;UPDATE POINTER TO NEXT CYLINDER BOUNDRY
                                                                 ;IN THE BSF TABLE.
 015202  005710                 2$:     TST     (R0)            ;IS THIS THE TERMINATOR OF THE BSF ?
 015204  100414                         BMI     4$              ;BR IF YES (SECTOR NOT FOUND)
 015206  020027  032462                 CMP     R0,#ENDBSF      ;IS THIS THE END OF THE BSF ENTRIES ?
 015212  101011                         BHI     4$              ;BR IF YES (SECTOR NOT FOUND)
 015214  023720  031460                 CMP     TMPBSF,(R0)+    ;SAME CYLINDER ADDRESS ?
 015220  001367                         BNE     1$              ;BR IF NO
 015222  023720  031462                 CMP     TMPBSF+2,(R0)+  ;SAME TRK/SEC ADDRESS ?
 015226  001365                         BNE     2$              ;BR IF NO
 015230  062766  000002  000002 3$:     ADD     #2,2(SP)        ;ADJUST RETURN FOR SECTOR FOUND
 015236                         4$:
 015236  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 015240  000207                         RTS     PC
```

```
                                .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
                                ;;*****************************************************************************

015242  005001                  MERGE:: CLR     R1              ;SET THE TOGGLE SWITCH TO NON-INTERLEAVED
015244  012705  033614                  MOV     #TDSBLK,R5      ;GET THE INITIAL TD TABLE ADDRESS,
015250  012700  002323                  MOV     #PLTRK,R0       ; THE RP07 INTERLEAVE MASK,
015254  112003                        MOVB    (R0)+,R3        ; THE 1ST INTERLEAVED SECTOR ADDRESS,
015256  111004                        MOVB    (R0),R4         ; THE 1ST SECTOR ADDRESS,
015260  012700  032464                MOV     #HDRBLK,R0      ; THE HEADER TABLE ADDRESS AND
015264  013737  002364  002232        MOV     TKWCNT,ENDTAB   ; THE # OF WORDS PER TRACK (6 WRDS X 50. SECTORS).
015272  006337  002232                ASL     ENDTAB          ;ADJUST WORDS TO BYTES (X 2) AND
015276  062737  032464  002232        ADD     #HDRBLK,ENDTAB  ;CALULATE END OF HEADER TABLE.
015304  012705  033614                MOV     #TDSBLK,R5      ;GET POINTER TO BEGINNING OF TD BLOCK
015310  053704  002314          1$:   BIS     DESTRK,R4       ;ADD THE TRACK ADDRESS TO
015314  053703  002314                BIS     DESTRK,R3       ;BOTH ADDRESS COUNTERS.
015320  013720  002312          2$:   MOV     DESCYL,(R0)+    ;LOAD THE CYLINDER ADDRESS
015324  005701                        TST     R1              ;SEE IF INTERLEAVED SECTOR SHOULD BE LOADED,
015326  001404                        BEQ     3$              ;BRANCH IF NOT, ELSE
015330  010310                        MOV     R3,(R0)         ;LOAD INTERLEAVED SECTOR,
015332  005203                        INC     R3              ;INCREMENT TO NEXT SECTOR ADDRESS,
015334  005001                        CLR     R1              ;SET THE TOGGLE SWITCH TO NON-INTERLEAVED AND
015336  000406                        BR      4$              ;CONTINUE.

015340  010410            3$:   MOV     R4,(R0)         ;LOAD LOW OR NON-INTERLEAVED SECTOR,
015342  005204                  INC     R4              ;INCREMENT TO NEXT SECTOR ADDRESS.
015344  105737  002244          TSTB    INTLEV          ;IS THE INTERLEAVE SWITCH ON?
015350  001401                  BEQ     4$              ;BR IF NO
015352  005101                  COM     R1              ;SET THE TOGGLE SWITCH TO INTERLEAVED
015354  011037  031462    4$:   MOV     (R0),TMPBSF+2   ;SETUP THE TRACK/SECTOR,
015360  014037  031460          MOV     -(R0),TMPBSF    ;CYLINDER TO BE CHECKED FOR IN THE BSF.
015364  004737  015172          JSR     PC,CHKBSF       ;NOW CHECK THE BAD SECTOR FILE,
015370  000403                  BR      5$              ;RETURN HERE IF SECTOR NOT FOUND,
                                                        ;ELSE RETURN HERE.
015372  052720  050000          BIS     #050000,(R0)+   ;SET UF/FMT BITS IN CYLINDER ADDRESS
015376  000402                  BR      6$

015400  052720  150000    5$:   BIS     #150000,(R0)+   ;SET MF/UF/FMT BITS IN CYLINDER ADDRESS
015404  005720            6$:   TST     (R0)+           ;GET POINTER OVER TRK/SEC ENTRY
015406  012520                  MOV     (R5)+,(R0)+     ;LOG THE TD DATA
015410  012520                  MOV     (R5)+,(R0)+     ;ALL FOUR WORDS OF IT
015412  012520                  MOV     (R5)+,(R0)+     ;FOR EACH SECTOR
015414  012520                  MOV     (R5)+,(R0)+     ;THIS IS LAST WORD FOR EACH SECTOR
015416  023700  002232          CMP     ENDTAB,R0       ;IS THIS THE END OF THE HEADER TABLE ?
015422  101412                  BLOS    8$              ;BR F YES
015424  123704  002322          CMPB    SEC50,R4        ;DONE WITH ALL SECTORS ON THIS TRACK ?
015430  103002                  BHIS    7$              ;BR IF NO
015432  005004                  CLR     R4              ;RESET R4 TO SECTOR ADDRESS 0 AND
```

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

```
                                .SBTTL  MAKE TRACK BUFFER

                                ;;*****************************************************************************
                                ;THIS MODULE IS USED ANYTIME A FORMAT TRACK OR VERIFY TRACK COMMAND IS ISSUED.
                                ;THE MODULE CREATES THE BUFFER NEED 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, BUSADR, NEGWRD, PPOF
                                ;*
                                ;CALL
                                ;       JSR     PC,MAKTRK       ;NOW EXECUTE THE COMMAND
                                ;       ------                  ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                ;       ------                  ;ELSE RETURN HERE
                                ;*
                                ;THIS MODULE IS CALLED BY: MAIN
                                ;;*****************************************************************************

015452                          MAKTRK:
015452  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
015454  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
015456  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
015460  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
015462  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
015464  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
015466  004737  014326                  JSR     PC,CALCTD       ;DO THE DEFECT LOCATING NOW
015472  004737  015242                  JSR     PC,MERGE        ;NOW PUT THE DEFECTS IN THE CORRECT SECTOR
015476  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
015500  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
015502  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
015504  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
015506  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
015510  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
015512  022737  000063  002404  1$:     CMP     @FMTRK,FUNCTN   ;IS THIS A FORMAT COMMAND?
015520  001004                          BNE     2$              ;BR IF NO
015522  012737  005700  002260          MOV     @FORMAT,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
015530  000413                          BR      3$              ;AND SKIP NEXT

015532  022737  000073  002404  2$:     CMP     @RDMD,FUNCTN    ;IS THIS A VERIFY FUNCTION?
015540  001007                          BNE     3$              ;BR IF NO (MUST BE A WRITE CHECK)
015542  012737  005713  002260          MOV     @RMDATA,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
015550  012737  033614  002234          MOV     @TDSBLK,BUSADR  ;GET POINTER TO BEGINNING OF TD BLOCK
015556  000403                          BR      4$              ;AND GO-ON

015560  012737  032464  002234  3$:     MOV     @HDRBLK,BUSADR  ;GET POINTER TO BEGINNING OF HEADER BLOCK
015566  013737  002364  002376  4$:     MOV     TKWCNT,NEGWRD   ;SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
015574  052737  100000  002236          BIS     @CMOD,OFFSET    ;SET COMMAND MODIFIER BIT
015602  004737  010502                  JSR     PC,DRIVER       ;EXECUTE THE OPERATION NOW
015606  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
015612  000405                          BR      5$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
015614  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
015620  001334                          BNE     1$              ;BR IF YES
015622  062716  000002                  ADD     #2,(SP)         ;ADJUST FOR GOOD RETURN
015626  000207                  5$:     RTS     PC              ;TAKE RETURN NOW
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE IS USED TO INITIALIZE THE RP07.  IT DETERMINES IF THE USER
 4                                ;SELECTED DRIVE IS ACTUALLY AN RP07.  IT ALSO CHECKS THE FUNCTIONALITY OF THE
 5                                ;'DRIVE READY' BIT IN RPDS.  IT DETERMINES IF THE DRIVE IS INTERLEAVE ENABLED
 6                                ;AND RECORDS THE STATUS FOR PROGRAM OPERATION.  IN ADDITION, IF A WRITE
 7                                ;OPERATION WAS SELECTED BY THE USER, THIS MODULE CHECKS THAT WRL IS NOT SET.
 8                                ;MOL MUST ALSO BE SET IN ORDER TO CORRECTLY COMPLETE THE DRIVE INITIALIZATION.
 9                                ;;*****************************************************************************
10
11 015630  004737  013256          DRVINI: JSR     PC,RESET        ;SEIZE THE PORT AND RESET THE MASSBUS
12                                                                 ;PRINT 'DRIVE X'
13 015634  013746  002434                  MOV     DRVNO,-(SP)
   015640  012746  003421                  MOV     #FRMT10,-(SP)
   015644  012746  000002                  MOV     #2,-(SP)
   015650  010600                          MOV     SP,R0
   015652  104417                          TRAP    C$PNTF
   015654  062706  000006                  ADD     #6,SP
14 015660  005777  164566                  TST     @RPDS           ;DO A READ OF THE DRIVE STATUS REG
15 015664  032777  010000  164556         BIT     #NED,@RPCS2     ;DO WE HAVE NED?
16 015672  001405                         BEQ     1$              ;IF ZERO, DRIVE DOES EXIST
17                                                                 ;REPORT 'NON-EXISTENT DRIVE'
18 015674  104455                         TRAP    C$ERDF
   015676  000001                         .WORD   1
   015700  007654                         .WORD   EM21
   015702  000000                         .WORD   0
19 015704  000572                         BR      NORUN           ;TAKE THE BAD BRANCH, THE DRIVE IS NO GOOD!
20
21 015706  032777  000004  164536  1$:    BIT     #ILEV,@RPDS     ;IS THIS DRIVE INTERLEAVE ENABLED?
22 015714  001414                         BEQ     2$              ;IF BIT = 0, NO
23 015716  112737  000377  002244         MOVB    #377,INTLEV     ;SET THE INTERLEAVE ENABLED MASK
24                                                                 ;PRINT ', INTERLEAVED'
25 015724  012746  004231                  MOV     #FRMT31,-(SP)
   015730  012746  000001                  MOV     #1,-(SP)
   015734  010600                          MOV     SP,R0
   015736  104417                          TRAP    C$PNTF
   015740  062706  000004                  ADD     #4,SP
26 015744  000410                         BR      3$
27 015746                          2$:                             ;PRINT ', NOT INTERLEAVED'
28 015746  012746  004203                  MOV     #FRMT30,-(SP)
   015752  012746  000001                  MOV     #1,-(SP)
   015756  010600                          MOV     SP,R0
   015760  104417                          TRAP    C$PNTF
   015762  062706  000004                  ADD     #4,SP
29 015766  017746  164474          3$:     MOV     @RPDT,-(SP)     ;GET THE DRIVE TYPE REGISTER
30 015772  042716  004000                  BIC     #BIT11,(SP)     ;CLEAR THE DRIVE REQUEST REQUIRED BIT
31 015776  022726  020042                  CMP     #20042,(SP)+    ;IS THE DRIVE AN RP07?
32 016002  001405                         BEQ     4$              ;BR IF YES
33                                                                 ;REPORT 'DRIVE NOT AN RP07'
34 016004  104455                         TRAP    C$ERDF
   016006  000002                         .WORD   2
   016010  007677                         .WORD   EM22
   016012  000000                         .WORD   0
35 016014  000526                         BR      NORUN
36
37 016016  012701  000004          4$:    MOV     #4,R1           ;GET AN OVERALL ITERATION COUNT
38 016022  105777  164424          5$:    TSTB    @RPDS           ;IS DRIVE READY SET ?
```

```
                                                BMI     6$              ;BR IF YES
                                                JSR     PC,WAIT         ;WAIT A LITTLE LONGER
                                                DEC     R1              ;DONE TRYING YET ?
                                                BGT     5$              ;BR IF NO
                                                MOV     #3,ERRMSK       ;LOG 'DRIVE HUNG, DRY NOT SET ERROR' STATUS
                                                MOV     #DRVRDY,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
                                                JSR     PC,ERRORS       ;REPORT THE ERROR
                                                BR      NORUN           ;DRIVE IS NO GOOD

                                        6$:     BIT     #MOL,@RPDS      ;IS THE DRIVE ONLINE?
                                                BNE     7$              ;BR IF YES
                                                MOV     #MOLINE,ERRFNC  ;MARK THE CURRENT FUNCTION
                                                MOV     #5,ERRMSK       ;LOG 'DRIVE OFFLINE ERROR' STATUS
                                                JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
                                                BR      NOGOOD          ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;ELSE RETURN HERE
                                                TST     RTYCNT          ;DID WE HAVE AN ERROR ?
                                                BNE     6$              ;BR IF YES

                                        7$:     TST     VRIFY           ;IS THIS VERIFY ONLY MODE ?
                                                BNE     9$              ;BR IF YES
                                                TST     LIST            ;IS THIS LIST MODE ?
                                                BNE     9$              ;BR IF YES
                                        8$:     BIT     #WRL,@RPDS      ;IS THE DRIVE WRITE LOCKED ?
                                                BEQ     9$              ;BR IF NO
                                                MOV     #WLOCK,ERRFNC   ;MARK THE WRITE LOCK TEST
                                                MOV     #4,ERRMSK       ;LOG 'DRIVE WRITE LOCKED ERROR' STATUS
                                                JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
                                                BR      NOGOOD          ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;ELSE RETURN HERE
                                                TST     RTYCNT          ;DID WE HAVE AN ERROR ?
                                                BNE     8$              ;BR IF YES

                                        9$:     MOV     #RECALI,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
                                                MOV     #RECAL,@RPCS1   ;DO RECALIBRATE COMMAND
                                                TSTB    @RPDS           ;DID DRIVE READY RESET DURING RECAL ?
                                                BPL     10$             ;BR IF YES
                                                MOV     #13,ERRMSK      ;LOG 'DRIVE READY DIDN'T RESET ERROR' STATUS
                                                JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
                                                BR      NOGOOD          ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;ELSE RETURN HERE
                                                TST     RTYCNT          ;DID WE HAVE AN ERROR ?
                                                BNE     9$              ;BR IF YES

                                        10$:    MOV     #RECAL,FUNCTN   ;SET UP FOR THE DRIVER
                                                JSR     PC,DRIVER       ;AND DO THE RECAL AGAIN
                                                JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
                                                BR      NOGOOD          ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;ELSE RETURN HERE
                                                TST     RTYCNT          ;DID WE HAVE AN ERROR ?
                                                BNE     10$             ;BR IF YES
                                                BR      BUFINI

                                        NORUN:  INC     ERRTTL          ;INCREMENT THE TOTAL ERROR COUNT
                                        NOGOOD: MOV     #-1,SOFSW       ;LOAD THE FAILED STATUS
                                        BUFINI: MOV     #BEGBUF,R1      ;GET POINTER TO BEGINNING OF BUFFER SECTION
                                        1$:     CLR     (R1)+           ;INITIALIZE ALL BUFFERS
```

M6


```
     96 016312  020127  120072                CMP     R1,#ENDBUF      ;DONE YET ?
     97 016316  101774                      BLOS    1$              ;BR IF NO
     98 016320  000207                      RTS     PC
```

```
 1
 2                              ;;*****************************************************************************
 3                              ;THIS MODULE CREATES THE TABLES WHICH ARE USED TO SELECT DEVICE PARAMETERS
 4                              ;USED IN THIS PROGRAM.  INCLUDED IN THE PARAMETER LIST ARE THE FOLLOWING ITEMS:
 5                              ;BASE REGISTER ADDRESSES FOR THE DEVICE, VECTOR ADDRESS OF THE DEVICE,
 6                              ;DETERMINATION VIA OPERATOR FOR THE FOLLOWING: CONTROLLER TYPE.  IN ADDITION,
 7                              ;DIALOGUE DETERMINES THE PRIORITY OF THE CONTROLLER, AND THE SELECTION OF A
 8                              ;DRIVE NUMBER FOR TEST.
 9                              ;*
10                              ;INPUTS ARE:    NON
11                              ;
12                              ;OUTPUTS ARE:   RPADD, RHTYPE, RPBAE, RPCS3, DRIVE, RPCS1
13                              ;*
14                              ;THIS MODULE IS CALLED BY: THE INITIALIZATION CODE
15                              ;;*****************************************************************************
16
17 016322                      TABELD::
   016322  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   016324  010346                      MOV     R3,-(SP)        ;;PUSH R3 ON STACK
   016326  010546                      MOV     R5,-(SP)        ;;PUSH R5 ON STACK
18 016330  012702  000024              MOV     #20.,R2         ;R2 = ITERATION COUNT
19 016334  012703  002440              MOV     #RPCS1,R3       ;R3 = DATA SINK
20 016340                      1$:                             ;GET HARDWARE P-TABLE
21 016340  013700  002420              MOV     UNIT,R0
   016344  104442                      TRAP    C$GPHRD
   016346  010005                      MOV     R0,R5
22 016350  103034                      BCC     4$
23 016352  011346                      MOV     (R3),-(SP)      ;SAVE R3
24 016354  011546                      MOV     (R5),-(SP)      ;AND THE BASE ADDRESS
25 016356  166616  000002              SUB     2(SP),(SP)      ;DERIVE NEW ADDRESS
26 016362  061623              2$:     ADD     (SP),(R3)+      ;LOG IT IN NEW TABLE
27 016364  005302                      DEC     R2              ;COUNT LOGGING
28 016366  001375                      BNE     2$              ;R2 NOT ZERO, CONTINUE LOGGING
29 016370  004737  010330              JSR     PC,SIZE70       ;SEE IF RH70 IS PRESENT
30 016374  005737  002432              TST     RHTYPE          ;IS IT AN RH70 ?
31 016400  001406                      BEQ     3$              ;BR IF NO
32 016402  013702  002430              MOV     RHEXT,R2        ;GET RPBAE OFFSET
33 016406  061502                      ADD     (R5),R2         ;ADD BASE ADDRESS TO OFFSET
34 016410  010223                      MOV     R2,(R3)+        ;SAVE NEW RPBAE
35 016412  005722                      TST     (R2)+           ;ADD 2
36 016414  010213                      MOV     R2,(R3)         ;SAVE NEW RPCS3
37
38 016416  022626              3$:     CMP     (SP)+,(SP)+     ;DONE, RESTORE THE STACK
39 016420  012537  002422              MOV     (R5)+,RPADR     ;SAVE RPCS1 BASE ADDRESS
40 016424  012537  002424              MOV     (R5)+,RPVEC     ;SAVE INTERUPT VECTOR ADDRESS
41 016430  012537  002426              MOV     (R5)+,RPVEC+2   ;SAVE INTERUPT PRIORITY
42 016434  011537  002434              MOV     (R5),DRVNO      ;SETUP DRIVE NUMBER FOR UNIT N
43 016440  000406                      BR      5$              ;SKIP NEXT
44
45 016442  005237  002420      4$:     INC     UNIT            ;TRY THE NEXT UNIT
46 016446  023737  002420  002012      CMP     UNIT,L$UNIT     ;IS THIS THE LAST UNIT TO TRY ?
47 016454  002731                      BLT     1$              ;BR IF NO
48 016456                      5$:
   016456  012605                      MOV     (SP)+,R5        ;;POP STACK INTO R5
   016460  012603                      MOV     (SP)+,R3        ;;POP STACK INTO R3
   016462  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
49 016464  000207                      RTS     PC              ;TAKE THE RETURN
```

```
                                .SBTTL  MAP SECTOR POSITION

                                ;;**********************************************************************
                                ;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
                                ;;**********************************************************************

016466  105737  002244      MAPSEC: TSTB    INTLEV          ;INTERLEAVED SECTORS ?
016472  001420                  BEQ     2$              ;BR IF NO
016474  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016476  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
016500  113701  002323          MOVB    PLTRK,R1        ;GET THE FIRST INTERLEAVED SECTOR
016504  013702  002402          MOV     SECADD,R2       ;SAVE THE SECTOR ADDRESS
016510  006301                  ASL     R1              ;MULTIPLY BY TWO
016512  006302                  ASL     R2              ;FOR THE THRESHOLD AND THE INPUT
016514  020201                  CMP     R2,R1           ;SCALE THE USER INPUT
016516  103402                  BLO     1$              ;IT JUST NEEDS DOUBLING, IF LOWER!
016520  162702  000061          SUB     #49.,R2         ;THIS IS A SECTOR ADDRESS => 25.
016524  010237  002402      1$:     MOV     R2,SECADD       ;SAVE THE CORRECTED SECTOR ADDRESS
016530  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
016532  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
016534  000207              2$:     RTS     PC              ;RETURN NOW!
```

```
                                .SBTTL  FIND DEFECT RELATIVE TO INDEX

                                ;;*****************************************************************************
                                ;THIS MODULE ACCEPTS SECTOR ADR AND WORD COUNT INPUT REPRESENTING A RELATIVE
                                ;DEFECT, WHICH IS CONVERTED TO ABSOLUTE DATA, RELATIVE ONLY TO INDEX.
                                ;*
                                ;INPUTS ARE:    INTLEV, SECADD, RELWRD
                                ;
                                ;OUTPUTS ARE:   RELWRD
                                ;*
                                ;THIS MODULE IS CALLED BY: INPUTD, HDSCAN, DASCAN
                                ;;*****************************************************************************

                                RELTIVE:
016536  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
016540  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016542  013746  002402                  MOV     SECADD,-(SP)    ;;PUSH SECADD ON STACK
016546  004737  016466                  JSR     PC,MAPSEC       ;CORRECT THE SECTOR MAP NOW!
016552  013700  002400                  MOV     RELWRD,R0       ;GET THE USER INPUT TO MAKE IT RELATIVE
016556  062700  000225                  ADD     #<G1+TD+G2+HDR0+G3>,R0  ;GET INFO TO PUT THE DEFECT
                                                                ;AT LEAST IN SECTOR 0
016562  012701  000517                  MOV     #<HDR+G3+DATA+G4>,R1    ;GET SIZE OF SECTOR
016566  005337  002402          1$:     DEC     SECADD          ;KEEP TABS OF THE ITERATION COUNT
016572  100402                          BMI     2$              ;WHEN MINUS, END!
016574  060100                          ADD     R1,R0           ;ADD ONE MORE SECTOR TO VALUE
016576  000773                          BR      1$              ;AND HOUSEKEEP

016600  010037  002400          2$:     MOV     R0,RELWRD       ;LOG THE ABSOLUTE VALUE RELATIVE TO INDEX
016604  012637  002402                  MOV     (SP)+,SECADD    ;;POP STACK INTO SECADD
016610  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
016612  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
016614  000207                          RTS     PC
```

```
                                .SBTTL  INSERT ABSOLUTE DEFECT INTO TDR

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

016616                          INSERT:
016616  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
016620  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
016622  012700  000004                  MOV     #4,R0           ;GET THE # OF DEFECT WORDS TO CHECK IN TD
016626  012701  031420                  MOV     #TDWRD3,R1      ;FIRST ENTRY OF RECORD
016632  022127  140000          1$:     CMP     (R1)+,#NULL     ;IS THE TRACK DESCRIPTOR FULL ?
016636  001403                          BEQ     2$              ;BR IF NO
016640  005300                          DEC     R0              ;ONE LESS AVAILABLE ENTRY TO-GO
016642  003373                          BGT     1$              ;LOOK AGAIN
016644  000430                          BR      4$              ;NO ROOM AT THE INN - WE'VE GOT TROUBLE.

016646  012737  177777  031460  2$:     MOV     #-1,TMPBSF      ;FLAG THAT DATA IS FOR TD, NOT BSF
016654  005737  031420                  TST     TDWRD3          ;IS DATA VALID ?
016660  001564                          BEQ     13$             ;BR IF NO
016662  012700  000004                  MOV     #4,R0           ;GET THE # OF DEFECT WORDS TO CHECK IN TD
016666  012701  031420                  MOV     #TDWRD3,R1      ;GET POINTER TO BEGINNING OF DEFECTS
016672  062737  000066  002400          ADD     #<DS/2>,RELWRD  ;ADD HALF A DEFECT TO ABSOLUTE VALUE
016700  023721  002400          3$:     CMP     RELWRD,(R1)+    ;DOES NEW DEFECT GO BEFORE THIS OLD DEFECT ?
016704  103513                          BLO     9$              ;BR IF YES
016706  166137  177776  002400          SUB     -2(R1),RELWRD   ;MAKE NEW DEFECT RELATIVE TO PREVIOUS DEFECT
016714  062737  000154  002400          ADD     #<DS>,RELWRD    ;ADD A DEFECT SKIP TO THE NEW DEFECT
016722  005300                          DEC     R0              ;DONE LOOKING YET ?
016724  003365                          BGT     3$              ;BR IF NO
                                                                ;SETUP TO PUT THIS NEW DEFECT INTO THE
                                                                ;BAD SECTOR FILE BUFFER.
016726  013737  002312  031460  4$:     MOV     DESCYL,TMPBSF   ;GET CYLINDER ADDRESS.
016734  013737  002402  031462          MOV     SECADD,TMPBSF+2 ;SECTOR ADDRESS AND
016742  113737  002315  031463          MOVB    DESTRK+1,TMPBSF+3       ;TRACK ADDRESS.
016750  113700  002315                  MOVB    DESTRK+1,R0     ;GET TRACK ADDRESS
                                                                ;PRINT 'WARNING ** CYL XXX., TRK XX. HAS 4
                                                                ;       DEFECTS IN TRACK DESCRIPTOR'

016754  010046                          MOV     R0,-(SP)
016756  013746  002312                  MOV     DESCYL,-(SP)
016762  012746  006206                  MOV     #WARN,-(SP)
016766  012746  005055                  MOV     #FRMT61,-(SP)
016772  012746  000004                  MOV     #4,-(SP)
016776  010600                          MOV     SP,R0
017000  104417                          TRAP    C$PNTF
017002  062706  000012                  ADD     #12,SP
017006  023737  031460  002222          CMP     TMPBSF,LSCYL    ;SEE WHAT CYLINDER ADDRESS IS BEING FLAGGED.
017014  101013                          BHI     6$              ;BRANCH IF FE CYLINDER(S).
017016  103404                          BLO     5$              ;BRANCH IF ANY USER CYLINDER, ELSE
```

INSERT ABSOLUTE DEFECT INTO TDR

```
                                                CMPB    TMPBSF+3,LSTRK  ;SEE IF WE ARE FLAGGING THE BSF AREA,
                                                BEQ     6$              ;BRANCH IF SO, ELSE
                                        5$:     TST     SCANR           ;SEE IF SCAN MODE IS ENABLED,
                                                BNE     7$              ;BRANCH IF SO, ELSE
                                                TST     MODBY           ;MUST BE MODIFY MODE, SO CHECK TO SEE IF MODIFIING
                                                                        ;BY WORD COUNT OR BY INDEX MODE,
                                                BEQ     8$              ;BRANCH IF WORD COUNT MODE, ELSE
                                        6$:                             ;PRINT '            INPUT DATA REJECTED'
                                                MOV     #FRMT60,-(SP)
                                                MOV     #1,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTF
                                                ADD     #4,SP
                                                BR      13$
                                        7$:                             ;PRINT '            ANOTHER DEFECT WAS FOUND
                                                                        ;       XXXXX. WORDS FROM INDEX'
                                                MOV     RELWRD,-(SP)
                                                MOV     #FRMT65,-(SP)
                                                MOV     #2,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTF
                                                ADD     #6,SP
                                        8$:                             ;PRINT '            BAD SECTOR FILE WILL BE UPDATED'
                                                MOV     #FRMT66,-(SP)
                                                MOV     #1,-(SP)
                                                MOV     SP,R0
                                                TRAP    C$PNTF
                                                ADD     #4,SP
                                                BR      13$

                        002400  9$:     SUB     #<DS/2>,RELWRD  ;SUBTRACT A HALF DEFECT FROM ABSOLUTE VALUE
                                        TST     -(R1)           ;BACK THE POINTER UP BY ONE WORD
                000001                  CMP     R0,#1           ;INSERT AFTER THREE DEFECTS?
                                        BEQ     12$             ;IF MATCH, YES
                000002                  CMP     R0,#2           ;INSERT AFTER TWO DEFECTS?
                                        BEQ     11$             ;IF MATCH, YES
                000003                  CMP     R0,#3           ;INSERT AFTER ONE DEFECT?
                                        BEQ     10$             ;IF MATCH, YES
                000004  000006          MOV     4(R1),6(R1)     ;SHUFFLE THE DATA DOWN
                000002  000004  10$:    MOV     2(R1),4(R1)     ;TO MAKE ROOM FOR THE NEW DEFECT
                000002          11$:    MOV     (R1),2(R1)      ;KEEP SHUFFLING DOWN
                002400          12$:    MOV     RELWRD,(R1)     ;LOG THE NEW DEFECT DATA NOW!
                                        DEC     R0              ;WAS THIS LAST ENTRY IN TD ?
                                        BEQ     13$             ;BR IF YES
                000002  140000          CMP     2(R1),#NULL     ;NEXT WORD NULL ?
                                        BEQ     13$             ;BR IF YES
                000002                  SUB     (R1),2(R1)      ;ADJUST NEXT DEFECT TO MAKE IT RELATIVE TO
                                                                ;THE NEW DEFECT
                                13$:
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        RTS     PC              ;NOW RETURN TO MAIN
```

```
                                .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
                                ;;*****************************************************************************

017240                          TDDUMP:
017240  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
017242  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
017244  005737  002172                  TST     LIST            ;IS LIST OPTION ENABLED ?
017250  001404                          BEQ     1$              ;BR IF NO
017252  023727  002254  000001          CMP     DEFCNT,#1       ;WAS FIRST DEFECT LISTED ALL READY ?
017260  002022                          BGE     2$              ;BR IF YES
017262                          1$:                             ;PRINT THE APPROPRIATE MESSAGE
017262  013746  002360                  MOV     TEXT,-(SP)
017266  012746  002661                  MOV     #FRMTT,-(SP)
017272  012746  000002                  MOV     #2,-(SP)
017276  010600                          MOV     SP,R0
017300  104417                          TRAP    C$PNTF
017302  062706  000006                  ADD     #6,SP
                                                                ;PRINT 'WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3 WORD#4 WORD#5
WORD#6'
017306  012746  004406                  MOV     #FRMT35,-(SP)
017312  012746  000001                  MOV     #1,-(SP)
017316  010600                          MOV     SP,R0
017320  104417                          TRAP    C$PNTF
017322  062706  000004                  ADD     #4,SP
017326  013746  031416          2$:     MOV     TDWRD2,-(SP)    ;GET THE 2ND WORD OF THE TD (TRK/SEC ADRS)
017332  011646                          MOV     (SP),-(SP)      ;SAVE 2ND TD WORD
017334  105066  000003                  CLRB    3(SP)           ;GET RID OF HI BYTE (EXTRACT SECTOR)
017340  011646                          MOV     (SP),-(SP)      ;SAVE 2ND TD WORD
017342  105066  000002                  CLRB    2(SP)           ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
017346  000366  000002                  SWAB    2(SP)           ;MAKE IT THE LOW BYTE DATA.
017352  013746  031414                  MOV     TDWRD1,-(SP)    ;GET THE FIRST WORD OF TD (CYLINDER ADDRESS)
017356  011646                          MOV     (SP),-(SP)      ;SAVE 1ST TD WORD
017360  042766  170000  000002          BIC     #170000,2(SP)   ;STRIP THE STATUS BITS
                                                                ;PRINT TD WORD#1, CYL, TD WORD#2 & TRK/SEC DATA
017366  012646                          MOV     (SP)+,-(SP)
017370  012646                          MOV     (SP)+,-(SP)
017372  012646                          MOV     (SP)+,-(SP)
017374  012646                          MOV     (SP)+,-(SP)
017376  012646                          MOV     (SP)+,-(SP)
017400  012746  004515                  MOV     #FRMT36,-(SP)
017404  012746  000006                  MOV     #6,-(SP)
017410  010600                          MOV     SP,R0
017412  104417                          TRAP    C$PNTF
017414  062706  000016                  ADD     #16,SP
017420  012701  031420                  MOV     #TDWRD3,R1      ;GET THE DEFECT FILE
017424  012702  000004                  MOV     #4,R2           ;AND THE MAXIMUM NUMBER OF DEFECTS
017430  022127  140000          3$:     CMP     (R1)+,#NULL     ;IS IT NULL DATA ?
017434  001413                          BEQ     4$              ;BR IF YES
```

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

```
                                .SBTTL  REGISTER BUFFER MODULE
                                
                                ;;*****************************************************************************
                                ;THIS MODULE SAVE AN IMAGE OF THE RP REGISTERS. IT IS USED FOR DIAGNOSTIC
                                ;PURPOSES SUBSEQUENT TO A DISK COMMAND.
                                ;*
                                ;INPUTS ARE:    RPCS1 - RPCS3.
                                ;
                                ;OUTPUTS ARE:   REG+0 - REG+52
                                ;*
                                ;THIS MODULE IS CALLED BY: ERRO
                                ;;*****************************************************************************
                                
017542                          SAVRPR:
017542  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
017544  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
017546  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
017550  012702  002440                  MOV     #RPCS1,R2       ;AND THE TRANSFER ADDRESSES
017554  012701  002514                  MOV     #REG,R1         ;OF THE SOURCE AND SINK BUFFERS
017560  012703  000024                  MOV     #20.,R3         ;GET THE ITERATION COUNT
017564  013221                  1$:     MOV     @(R2)+,(R1)+    ;NOW LOG THE DATA
017566  005303                          DEC     R3              ;ONE LESS REGISTER TO GO
017570  003375                          BGT     1$              ;IF >0, KEEP GOING
017572  005737  002432                  TST     RHTYPE          ;WHICH CONTROLLER?
017576  001402                          BEQ     2$              ;IF ZERO, IT'S AN RH11
017600  013221                          MOV     @(R2)+,(R1)+    ;LOG RPBAE
017602  013221                          MOV     @(R2)+,(R1)+    ;LOG RPCS3
017604                          2$:
017604  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
017606  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
017610  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
017612  000207                          RTS     PC              ;RETURN TO MAIN FOR ERROR REPORT
```

HEADER VERIFY MODULE

```
                                .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
                                ;;*****************************************************************************

017614  012700  032464        VERIFY: MOV     #HDRBLK,R0      ;GET POINTER TO BEGINNING OF EXPECTED DATA
017620  105737  002244                TSTB    INTLEV          ;INTERLEAVED DRIVE ?
017624  001404                        BEQ     1$              ;BR IF NO
017626  004737  011630                JSR     PC,CHABUF       ;RE-SHUFFLE THE DATA FOR A LINEAR COMPARE
017632  012700  034744                MOV     #LINBUF,R0      ;GET POINTER TO BEGINNING OF EXPECTED DATA
017636  113702  002322        1$:     MOVB    SEC50,R2        ;# OF SECTORS ON TRACK
017642  012701  033614                MOV     #TDSBLK,R1      ;GET POINTER TO BEGINNING OF RECEIVED DATA
017646  012704  000006        2$:     MOV     #<HDR-2>,R4     ;# OF WORDS IN HEADER TO VERIFY
017652  022021                3$:     CMP     (R0)+,(R1)+     ;DOES EXPECTED DATA MATCH RECEIVED DATA ?
017654  001007                        BNE     4$              ;BR IF NO
017656  005304                        DEC     R4              ;DONE COMPARING THIS SECTOR YET ?
017660  001374                        BNE     3$              ;BR IF NO
017662  005302                        DEC     R2              ;DONE COMPARING THIS TRACK YET ?
017664  002370                        BGE     2$              ;BR IF NO
017666  005037  002272                CLR     ERRMSK          ;RESET NO ERROR STATUS AND SETUP CONDITION CODE
                                                              ;IN PSW FOR RETURN
017672  000207                        RTS     PC

017674  014037  002276        4$:     MOV     -(R0),EXPCTD    ;SAVE THE EXPECTED DATA
017700  014137  002300                MOV     -(R1),RECVED    ;AND THE RECEIVED DATA
017704  012737  000010  002272        MOV     #10,ERRMSK      ;LOG 'FORMAT VERIFY ERROR' STATUS AND
                                                              ;SETUP CONDITION CODE IN PSW FOR RETURN
017712  000207                        RTS     PC
```

```
                                .SBTTL  READ MFG BAD SECTOR FILE MODULE
                                ;
                                ;;*****************************************************************************
                                ;THIS MODULE DOES A READ OF THE MANUFACTURES SECTION (SECTORS 0,2,4,6,8) OF
                                ;THE BAD SECTOR FILE (DEC144). IF THE READ IS UNSUCCESSFUL, THE PROGRAM
                                ;ASSUMES THAT THE BSF WAS NEVER CREATED AND MARKS STATUS AS SUCH. IF DATA IS
                                ;FOUND FROM ONE OF THE FIVE COPIES OF THE MFG'S BAD SECTOR FILE, THE
                                ;INFORMATION IS STORED FOR A LATER REWRITE.
                                ;*
                                ;INPUTS ARE:    MFGBUF, RDDAT, RTYCNT
                                ;
                                ;OUTPUTS ARE:   SUPRSS, DESCYL, DESTRK, BUSADR, FUNCTN, NEGWRD, ERRMAX, CREATE
                                ;*
                                ;THIS MODULE IS CALLED BY: MAIN
                                ;;*****************************************************************************

017714  012737  177777  002252  RDBSF:  MOV     #-1,SUPRSS      ;SUPRESS THE ERROR MESSAGE OUTPUT
017722  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
017724  013746  002312                  MOV     DESCYL,-(SP)    ;;PUSH DESCYL ON STACK
017730  013746  002314                  MOV     DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
017734  013746  002404                  MOV     FUNCTN,-(SP)    ;;PUSH FUNCTN ON STACK
017740  013737  002222  002312          MOV     LSCYL,DESCYL    ;GET THE BAD SECTOR FILE (LAST) CYLINDER,
017746  112737  000000  002314          MOVB    #0,DESTRK       ;SECTOR AND
017754  113737  002224  002315          MOVB    LSTRK,DESTRK+1  ;TRACK ADDRESSES.
017762  012737  031464  002234  1$:     MOV     #MFGBUF,BUSADR  ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
017770  012737  000071  002404          MOV     #RDDAT,FUNCTN   ;SET UP FOR A READ DATA COMMAND
017776  012737  000400  002376          MOV     #256.,NEGWRD    ;WORD COUNT FOR THE ENTIRE SECTOR
020004  004737  010502                  JSR     PC,DRIVER       ;EXECUTE THE OPERATION
020010  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
020014  000404                          BR      2$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
020016  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
020022  001357                          BNE     1$              ;BR IF YES
020024  000425                          BR      3$              ;GOOD READ!!!

020026  062737  000002  002314  2$:     ADD     #2,DESTRK       ;INCREMENT TO NEXT COPY OF MFG BSF
020034  123727  002314  000012          CMPB    DESTRK,#10.     ;DID WE TRY READING ALL COPIES OF MFG BSF ?
020042  002747                          BLT     1$              ;BR IF NO
                                                                ;PRINT 'WARNING ** BAD SECTOR FILE DATA IS
                                                                ;       CORRUPT, IT WILL BE INITIALIZED'
020044  012746  006206                  MOV     #WARN,-(SP)
020050  012746  005342                  MOV     #FRMT64,-(SP)
020054  012746  000002                  MOV     #2,-(SP)
020060  010600                          MOV     SP,R0
020062  104417                          TRAP    C$PNTF
020064  062706  000006                  ADD     #6,SP
020070  012700  031464                  MOV     #MFGBUF,R0      ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
020074  004737  020550                  JSR     PC,CRE144       ;CREATE A NULLED MFG BSF BUFFER
                                                                ;NOW THAT WE HAVE A GOOD READ OR AN INITIALIZED
                                                                ;BUFFER, LET'S SEE IF THE DATA IS VALID.
020100  012700  031464          3$:     MOV     #MFGBUF,R0      ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
020104  022027  000105                  CMP     (R0)+,#105      ;LOOK FOR THE PACK S.N.
020110  001346                          BNE     2$              ;NO MATCH, REWRITE THE MFG BSF
020112  022027  000105                  CMP     (R0)+,#105      ;LOOK FOR THE PACK S.N.
020116  001343                          BNE     2$              ;NO MATCH, REWRITE THE MFG BSF
020120  005720                          TST     (R0)+           ;THIS WORD ALWAYS 0!
020122  001341                          BNE     2$              ;IF NOT, FILE IS BAD
```

```
    50 020124  005720                    TST     (R0)+           ;IS THIS AN ALIGNMENT PACK ?
    51 020126  001337                    BNE     2$              ;BR IF YES
    52 020130  004737  013256            JSR     PC,RESET        ;RESET ANY ERRORS
    53 020134  012637  002404            MOV     (SP)+,FUNCTN    ;;POP STACK INTO FUNCTN
       020140  012637  002314            MOV     (SP)+,DESTRK    ;;POP STACK INTO DESTRK
       020144  012637  002312            MOV     (SP)+,DESCYL    ;;POP STACK INTO DESCYL
       020150  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
    54 020152  005037  002252            CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
    55 020156  000207                    RTS     PC              ;RETURN TO MAIN
```

```
                                .SBTTL  WRITE MFG/USR BAD SECTOR FILE MODULE
                                ;
                                ;;*****************************************************************************
                                ;THIS MODULE CREATES AND WRITES THE BAD SECTOR FILE ON THE LAST USER
                                ;CYLINDER/TRACK ON THE DISK.
                                ;*
                                ;INPUTS ARE:    WRTDAT, WRDAT, MFGBUF, DBUFF, CREATE
                                ;
                                ;OUTPUTS ARE:   DESCYL, DESTRK, FUNCTN, ERRFNC, MFG144, BUSADR, NEGWRD, ERRMSK,
                                ;               SEC50
                                ;*
                                ;THIS MODULE IS CALLED BY: MAIN
                                ;;*****************************************************************************
                                ;
020160                          WRTBSF:
020160  013746  002312                  MOV     DESCYL,-(SP)    ;;PUSH DESCYL ON STACK
020164  013746  002314                  MOV     DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
020170  004737  020414                  JSR     PC,SORT         ;SORT THE MFG BSF ENTRIES
020174  012700  036074                  MOV     #DBUFF,R0       ;GET POINTER TO BEGINNING OF BUFFER
020200  004737  020550                  JSR     PC,CRE144       ;CREATE THE USR BSF BUFFER
020204  013737  002222  002312          MOV     LSCYL,DESCYL    ;GET THE BAD SECTOR FILE (LAST) CYLINDER,
020212  112737  000000  002314          MOVB    #0,DESTRK       ;SECTOR AND
020220  113737  002224  002315          MOVB    LSTRK,DESTRK+1  ;TRACK ADDRESSES.
020226  012737  031464  002234  1$:     MOV     #MFGBUF,BUSADR  ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
020234  012737  000061  002404  2$:     MOV     #WRTDAT,FUNCTN  ;SET UP FOR A WRITE DATA FUNCTION
020242  012737  005625  002260          MOV     #WRDAT,ERRFNC   ;SAVE FUNCTION FOR ERROR ROUTINE
020250  012737  000400  002376          MOV     #256.,NEGWRD    ;AND THE SECTOR WORD COUNT
020256  004737  010502                  JSR     PC,DRIVER       ;EXECUTE THE OPERATION
020262  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
020266  000403                          BR      3$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
020270  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
020274  001357                          BNE     2$              ;BR IF YES
020276  105237  002314          3$:     INCB    DESTRK          ;NEXT SECTOR
020302  123727  002314  000012          CMPB    DESTRK,#10.     ;DONE WITH 16 & 18 BIT MFG BSF YET ?
020310  002010                          BGE     4$              ;BR IF YES
020312  022737  036074  002234          CMP     #DBUFF,BUSADR   ;DID WE JUST WRITE A 18 BIT MFG BSF ?
020320  001742                          BEQ     1$              ;BR IF YES
020322  012737  036074  002234          MOV     #DBUFF,BUSADR   ;GET POINTER TO BEGINNING OF 18 BIT BSF BUFFER
020330  000741                          BR      2$
                                                                ;START WRITTING USR 16 & 18 BIT BSF
020332  012737  036074  002234  4$:     MOV     #DBUFF,BUSADR   ;GET POINTER TO BEGINNING OF USR BSF BUFFER
020340  012737  000400  002376          MOV     #256.,NEGWRD    ;AND THE SECTOR WORD COUNT
020346  004737  010502                  JSR     PC,DRIVER       ;EXECUTE THE OPERATION
020352  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
020356  000403                          BR      5$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
020360  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
020364  001362                          BNE     4$              ;BR IF YES
020366  105237  002314          5$:     INCB    DESTRK          ;NEXT SECTOR
020372  123737  002314  002322          CMPB    DESTRK,SEC50    ;DONE WITH USR 16 & 18 BIT BSF YET ?
020400  003754                          BLE     4$              ;BR IF NO
020402  012637  002314                  MOV     (SP)+,DESTRK    ;;POP STACK INTO DESTRK
020406  012637  002312                  MOV     (SP)+,DESCYL    ;;POP STACK INTO DESCYL
020412  000207                          RTS     PC
```

```
                                .SBTTL  SORT BAD SECTOR FILE

                                ;;*****************************************************************************
                                ;THIS ROUTINE WILL SORT OUT DUPLICATE ENTRIES AND PUT THE BAD SECTOR FILE
                                ;IN ASCENDING ORDER.
                                ;CALL
                                ;       JSR     PC,SORT         ;CALL THE ROUTINE
                                ;;*****************************************************************************

020414  000240                  SORT:   NOP
020416  012701  031474                  MOV     #MFGBUF+8.,R1   ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
                                                                ;TO GET OVER 2 S/N WRDS AND 2 ALL 0'S WRDS.
020422  012703  032462                  MOV     #ENDBSF,R3      ;GET POINTER TO END OF MFG BSF BUFFER
020426  005711                          TST     (R1)            ;DONE CHECKING LIST YET ?
020430  100446                          BMI     5$              ;BR IF YES
020432  010102                  1$:     MOV     R1,R2           ;RESTORE BASE POINTER TO R2
020434  000422                          BR      3$
020436  021211                  2$:     CMP     (R2),(R1)       ;COMPARE THE CYLINDER NUMBER
020440  103427                          BLO     4$              ;SWAP ENTRIES
020442  101017                          BHI     3$              ;INCREMENT POINTERS
020444  126261  000003  000003          CMPB    3(R2),3(R1)     ;COMPARE THE TRACK NUMBER
020452  103422                          BLO     4$              ;SWAP ENTRIES
020454  101012                          BHI     3$              ;INCREMENT POINTERS
020456  126261  000002  000002          CMPB    2(R2),2(R1)     ;COMPARE THE SECTOR NUMBER
020464  103415                          BLO     4$              ;SWAP ENTRIES
020466  101005                          BHI     3$              ;INCREMENT POINTERS
020470  012712  177777                  MOV     #-1,(R2)        ;GET RID OF DUPLICATE CYLINDER AND
020474  012762  177777  000002          MOV     #-1,2(R2)       ;DUPLICATE TRACK/SECTOR ENTRIES.
020502  022222                  3$:     CMP     (R2)+,(R2)+     ;UPDATE SEARCH POINTER TO NEXT CYLINDER
020504  020203                          CMP     R2,R3           ;SEE IF THIS IS THE END OF BSF BUFFER.
020506  103753                          BLO     2$              ;BRANCH IF NOT, ELSE
020510  022121                          CMP     (R1)+,(R1)+     ;UPDATE BASE POINTER TO NEXT CYLINDER
020512  020103                          CMP     R1,R3           ;SEE IF THIS IS THE END OF BSF BUFFER.
020514  103746                          BLO     1$              ;BRANCH IF NOT, ELSE
020516  000413                          BR      5$              ;EXIT

020520  011146                  4$:     MOV     (R1),-(SP)      ;SAVE THE CURRENT LO CYLINDER ENTRY
020522  016146  000002                  MOV     2(R1),-(SP)     ;SAVE THE CURRENT LO TRACK/SECTOR ENTRY
020526  011211                          MOV     (R2),(R1)       ;SWITCH THE LO CYLINDER ENTRY
020530  016261  000002  000002          MOV     2(R2),2(R1)     ;SWITCH THE LO TRACK/SECTOR ENTRY
020536  012662  000002                  MOV     (SP)+,2(R2)     ;LOAD THE HI CYLINDER ENTRY
020542  012612                          MOV     (SP)+,(R2)      ;LOAD THE HI TRACK/SECTOR ENTRY
020544  000756                          BR      3$

020546  000207                  5$:     RTS     PC              ;EXIT
```

```
 1                                      ;;*****************************************************************************
 2                                      ;THIS MODULE CREATES THE MANUFACTURES OR USER SECTION OF THE BAD SECTOR FILE
 3                                      ;(DEC 144 FILE), WHICH IS WRITTEN TO THE LAST USER CYLINDER/TRACK ON THE DISK.
 4                                      ;*
 5                                      ;INPUTS ARE:    R0 = POINTER TO BEGINNING OUTPUT BUFFER AREA
 6                                      ;
 7                                      ;OUTPUTS ARE:   BUFFER POINTED TO BY R0
 8                                      ;*
 9                                      ;THIS MODULE IS CALLED BY: RDBSF, WRTBSF
10                                      ;;*****************************************************************************
11
12 020550  010001              CRE144: MOV     R0,R1           ;GET THE START OF THE BUFFER
13 020552  062701  001000              ADD     #<256.*2>,R1    ;SAVE END OF BUFFER IN R1
14 020556  012720  000105              MOV     #105,(R0)+      ;CREATE A "HDA SERIAL NUMBER"
15 020562  012720  000105              MOV     #105,(R0)+      ;TO SATISFY ALL OPERATING SYSTEMS
16 020566  005020                      CLR     (R0)+           ;THIRD WORD ALWAYS = 0!
17 020570  005020                      CLR     (R0)+           ;FOURTH WORD = 0 (<> ALIGNMENT PACK)
18 020572  012720  177777      1$:     MOV     #-1,(R0)+       ;NULL THE BAD SECTOR INFORMATION
19 020576  020001                      CMP     R0,R1           ;FILE DONE?
20 020600  103774                      BLO     1$              ;NOT IF R0 < SAID NUMBER
21 020602  000207                      RTS     PC              ;OK, TAKE RETURN
22
```

```
                                .SBTTL  ADDRESSING ROUTINES

                                ;;*****************************************************************************
                                ;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
                                ;;*****************************************************************************

020604  013701  002256        GETNEX: MOV     TBLPTR,R1       ;SETUP DEFECT BUFFER POINTER
020610  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
020612  012137  002312                MOV     (R1)+,DESCYL    ;GET THE CYLINDER ADDRESS
020616  042737  170000  002312        BIC     #170000,DESCYL  ;REMOVE ANY STATUS BITS
020624  012137  002314                MOV     (R1)+,DESTRK    ;AND LOAD IT NOW
020630  105037  002314                CLRB    DESTRK          ;EXTRACT SECTOR FROM DISK ADDRESS
020634  062737  000016  002256        ADD     #<7*2>,TBLPTR   ;GET NEXT ENTRY IN BUFFER
020642  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
020644  000207                1$:     RTS     PC

                                ;;*****************************************************************************
                                ;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.
                                ;
                                ;INPUTS ARE:    MINTRK, MAXTRK
                                ;
                                ;OUTPUTS ARE:   DESCYL, DESTRK
                                ;*
                                ;THIS MODULE IS CALLED BY: MAIN
                                ;;*****************************************************************************

020646  105237  002315        SPIRAL: INCB    DESTRK+1        ;INCREMENT TO NEXT TRACK ADDRESS
020652  123737  002315  002220        CMPB    DESTRK+1,MAXTRK ;TRACK COUNT AT MAX?
020660  101405                        BLOS    1$              ;BR IF NO
020662  113737  002216  002315        MOVB    MINTRK,DESTRK+1 ;RESTORE MIN TRACK ADDRESS
020670  005237  002312                INC     DESCYL          ;INCREMENT TO NEXT CYLINDER ADDRESS
020674  000207                1$:     RTS     PC              ;EXIT
```

```
                                ;;*****************************************************************************
                                ;THIS MODULE IS USED TO CONTROL THE COUNTING OF DEFECTS ENCOUNTERED ON THE
                                ;RP07 HDA.
                                ;*
                                ;INPUTS ARE:    FROM TDWRD3 - TDWRD6
                                ;
                                ;OUTPUTS ARE:   DEF1 - DEF4, TRACK AND DEFCNT (DEPENDENT UPON THE # OF DEFECTS
                                ;               FOUND)
                                ;*
                                ;CALL
                                ;       JSR     PC,CNTDEF
                                ;;*****************************************************************************

020676                          CNTDEF:
020676  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
020700  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
020702  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
020704  023727  031420  140000          CMP     TDWRD3,#NULL    ;DEFECT INFORMATION ?
020712  001452                          BEQ     3$              ;BR IF NO
020714  005737  031420                  TST     TDWRD3          ;IS IT VALID DATA ?
020720  001447                          BEQ     3$              ;BR IF NO
020722  012702  000004                  MOV     #4,R2           ;FOUR RECORDS
020726  012701  031420                  MOV     #TDWRD3,R1      ;INPUT FROM HERE
020732  012703  002302                  MOV     #DEF1,R3        ;OUTPUT COUNT GOES HERE
020736  022127  140000          1$:     CMP     (R1)+,#NULL     ;DATA OR UNUSED ?
020742  001435                          BEQ     2$              ;IF MATCH, UNUSED
020744  005237  002254                  INC     DEFCNT          ;INCLUDE THIS IN THE OVERALL COUNT
020750  005723                          TST     (R3)+           ;BUMP THE POINTER
020752  005302                          DEC     R2              ;ONE LESS ENTRY TO GO
020754  003370                          BGT     1$              ;IF NOT ZERO, WE'RE NOT DONE
020756  005737  002172                  TST     LIST            ;IS LIST OPTION ENABLED ?
020762  001025                          BNE     2$              ;IF NOT ZERO, YES
020764  005737  002170                  TST     MODTD           ;IS MODIFY TD MODE ENABLED ?
020770  001022                          BNE     2$              ;BR IF YES
020772  013701  031414                  MOV     TDWRD1,R1       ;GET THE 1ST WORD OF TD (CYLINDER ADDRESS)
020776  042701  170000                  BIC     #170000,R1      ;STRIP THE STATUS BITS
021002  113702  031417                  MOVB    TDWRD2+1,R2     ;GET HI BYTE OF 2ND WORD OF TD (TRACK ADDRESS)
                                        ;PRINT 'WARNING ** CYL XXX., TRK XX. HAS 4
                                        ;       DEFECTS IN TRACK DESCRIPTOR'
021006  010246                          MOV     R2,-(SP)
021010  010146                          MOV     R1,-(SP)
021012  012746  006206                  MOV     #WARN,-(SP)
021016  012746  005055                  MOV     #FRMT61,-(SP)
021022  012746  000004                  MOV     #4,-(SP)
021026  010600                          MOV     SP,R0
021030  104417                          TRAP    C$PNTF
021032  062706  000012                  ADD     #12,SP
021036  005243                  2$:     INC     -(R3)           ;INCLUDE THIS IN # DEFECTS/TRACK
021040                          3$:
021040  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
021042  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
021044  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
021046  000207                          RTS     PC              ;RETURN TO THE USER
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE IS USED ONLY TO RE-WRITE TD'S ON THE SECOND FE CYLINDER. THESE
 4                                ;TD'S ARE NECESSARY IN-ORDER FOR THE INTERNAL RP07 MICRODIAGNOSTICS TO
 5                                ;PROPERLY EXECUTE.
 6                                ;;*****************************************************************************
 7
 8 021050  012737  001167  002312 FETWO:  MOV     #631.,DESCYL     ;GET THE SECOND FE CYLINDER ADDRESS
 9 021056  005037  002314           CLR     DESTRK           ;RESET THE TRACK/SECTOR ADDRESS
10 021062  012701  031414           1$:     MOV     #TDWRD1,R1       ;SET UP A BUFFER POINTER
11 021066  013711  002312                   MOV     DESCYL,(R1)      ;FIRST WORD OF TD
12 021072  052721  030000                   BIS     #BIT13!BIT12,(R1)+       ;IT'S A TD - IN 16 BIT MODE
13 021076  013711  002314                   MOV     DESTRK,(R1)      ;TRACK / SECTOR ADDRESS
14 021102  052721  000377                   BIS     #377,(R1)+       ;SECTOR ADDRESS = - 1
15 021106  012702  000004                   MOV     #4,R2            ;A RECORD ITERATION COUNTER
16 021112  012721  140000           2$:     MOV     #NULL,(R1)+      ;NO DATA FOR THIS TD (LEAVE IT NULL)
17 021116  005302                   DEC     R2               ;ONE LESS RECORD TO GO
18 021120  003374                   BGT     2$               ;IF NOT ZERO, WE'RE NOT DONE
19 021122  004737  021152           JSR     PC,WRITTD        ;ATTEMPT TO WRITE THIS RECORD
20 021126  000207                   RTS     PC               ;RETURN HERE IF EXCEEDED RETRY LIMIT
21                                                            ;OTHERWISE RETURN HERE
22 021130  105237  002315                   INCB    DESTRK+1         ;NEXT TRACK PLEASE.....
23 021134  123727  002315  000037         CMPB    DESTRK+1,#31.    ;LAST TRACK YET?
24 021142  101747                   BLOS    1$               ;TAKE BRANCH IF NOT
25 021144  105337  002315                   DECB    DESTRK+1         ;GET RID OF THE OVERFLOW ADDRESS
26 021150  000207                   RTS     PC               ;TAKE THE USER RETURN
```

```
                                        .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
                                        ;BETWEEN 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, WRTTD, TDBCNT, RTYCNT
                                        ;
                                        ;OUTPUTS ARE:   DESTRK, RTYCNT, FUNCTN, BUSADR, ERRFNC, RPOF, NEGWRD, ERRMSK
                                        ;*
                                        ;CALL
                                        ;       JSR     PC,WRITTD       ;CALL WRITE TD SUBROUTINE
                                        ;       ------                  ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                        ;       ------                  ;ELSE RETURN HERE, NO ERROR
                                        ;*
                                        ;THIS MODULE IS MAINLY CALLED BY: INPUTD, FETWO, MAIN
                                        ;;*****************************************************************************

021152                                  WRITTD:
021152  010046                                MOV     R0,-(SP)        ;;PUSH R0 ON STACK
021154  010146                                MOV     R1,-(SP)        ;;PUSH R1 ON STACK
021156  013746  002314                        MOV     DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
021162  012700  002352                1$:     MOV     #TDLMTS,R0      ;GET POINTER TO TD LIMITS
021166  012701  031420                        MOV     #TDWRD3,R1      ;GET TRACK DESCRIPTOR POINTER
021172  112737  000377  002314                MOVB    #377,DESTRK     ;LOAD THE SECTOR ADDRESS BITS
021200  021127  140000                        CMP     (R1),#NULL      ;ANY DEFECT SKIPS ON THIS TRACK ?
021204  001414                                BEQ     2$              ;BR IF NO
021206  021120                                CMP     (R1),(R0)+      ;WAS TD MOVED ?
021210  003455                                BLE     5$              ;BR IF YES
021212  032737  040000  031414                BIT     #BIT14,TDWRD1   ;TD BIT ALREADY SET ?
021220  001051                                BNE     5$              ;BR IF YES
021222  021120                                CMP     (R1),(R0)+      ;WAS HDR 0 MOVED ?
021224  003443                                BLE     4$              ;BR IF YES
021226  032737  100000  031414                BIT     #BIT15,TDWRD1   ;HDR 0 BIT ALREADY SET ?
021234  001037                                BNE     4$              ;BR IF YES
021236  012737  000065  002404          2$:   MOV     #WRTTD,FUNCTN   ;SET UP FOR A WRITE TD FUNCTION
021244  012737  031414  002234                MOV     #TDWRD1,BUSADR  ;GET POINTER TO BEGINNING OF TRACK DESCRIPTOR BUFFER
021252  012737  005604  002260                MOV     #MSWRTD,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
021260  052737  100000  002236                BIS     #CMOD,OFFSET    ;SET COMMAND MODIFIER BIT
021266  013737  002330  002376                MOV     TDBCNT,NEGWRD   ;6 WORD TRANSFER
021274  004737  010502                        JSR     PC,DRIVER       ;SEND COMMAND TO CONTROLLER
021300  004737  013304                        JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
021304  000406                                BR      3$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;ELSE RETURN HERE
021306  005737  002274                        TST     RTYCNT          ;DID WE HAVE AN ERROR ?
021312  001323                                BNE     1$              ;BR IF YES
021314  062766  000002  000006                ADD     #2,6(SP)        ;ADJUST FOR GOOD RETURN
021322                                  3$:
021322  012637  002314                        MOV     (SP)+,DESTRK    ;;POP STACK INTO DESTRK
021326  012601                                MOV     (SP)+,R1        ;;POP STACK INTO R1
021330  012600                                MOV     (SP)+,R0        ;;POP STACK INTO R0
021332  000207                                RTS     PC              ;RETURN

021334  052737  100000  031414          4$:   BIS     #BIT15,TDWRD1   ;MARK HDR 0 MOVED
```

```
     52 021342  000735                              BR      2$              ;EXECUTE FUNCTION NOW
     53
     54 021344  052737  040000  002236  5$:         BIS     #MTD,OFFSET     ;MARK TD MOVED IN OFFSET REG
     55 021352  052737  040000  031414              BIS     #BIT14,TDWRD1   ;MARK TD MOVED IN TRACK DESCRIPTOR
     56 021360  026127  000002  140000              CMP     2(R1),#NULL     ;ANY MORE DEFECT SKIPS ON THIS TRACK ?
     57 021366  001723                              BEQ     2$              ;BR IF NO
     58 021370  032737  100000  031414              BIT     #BIT15,TDWRD1   ;HDR 0 BIT ALREADY SET ?
     59 021376  001356                              BNE     4$              ;BR IF YES
     60 021400  016146  000002                      MOV     2(R1),-(SP)     ;SAVE NEXT DEFECT VALUE
     61 021404  016046  177776                      MOV     -2(R0),-(SP)    ;GET UPPER LIMIT FOR TD MOVED
     62 021410  161116                              SUB     (R1),(SP)       ;FIND DIFFERENCE AND
     63 021412  162616                              SUB     (SP)+,(SP)      ;ALSO, TAKE ANY DIFFERENCE FROM NEXT DEFECT
     64 021414  016001  000002                      MOV     2(R0),R1        ;GET UPPER LIMIT FOR HDR 0 MOVED
     65 021420  022601                              CMP     (SP)+,R1        ;WAS HEADER 0 ALSO MOVED ?
     66 021422  003305                              BGT     2$              ;BR IF NO
     67 021424  005746                              TST     -(SP)           ;BACKUP TO GET VALUE AGAIN
     68 021426  162701  000066                      SUB     #<DS/2>,R1      ;SUBTRACT HALF A DEFECT TO GET LOWER LIMIT
     69                                                                     ;FOR HDR 0 MOVED
     70 021432  022601                              CMP     (SP)+,R1        ;WAS HEADER 0 ALSO MOVED ?
     71 021434  003700                              BLE     2$              ;BR IF NO
     72 021436  000736                              BR      4$
```

```
                                        .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, BUSADR, NEGWRD, TDCPY1
                                        ;*
                                        ;CALL
                                        ;       JSR     PC,READTD       ;CALL READ TD SUBROUTINE
                                        ;       ------                  ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                        ;       ------                  ;ELSE RETURN HERE, NO ERROR
                                        ;*
                                        ;THIS MODULE IS MAINLY CALLED BY: INPUTD, TDSCAN, UPDSCR, MAIN
                                        ;;**********************************************************************
                                        READTD:
021440  013746  002404                          MOV     FUNCTN,-(SP)    ;;PUSH FUNCTN ON STACK
021444  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
021446  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
021450  012737  000075  002404                  MOV     #RDTD,FUNCTN    ;ISSUE THE READ TD COMMAND
021456  012737  005615  002260                  MOV     #MSRDTD,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
021464  012700  031414              1$:         MOV     #TDWRD1,R0      ;GET POINTER TO BEGINNING OF TD BUFFER
021470  013701  002330                          MOV     TDBCNT,R1       ;GET BUFFER SIZE
021474  005020                      2$:         CLR     (R0)+           ;INITIALIZE THE BUFFER
021476  005301                                  DEC     R1              ;ONE LESS ENTRY TO GO
021500  003375                                  BGT     2$              ;IF NOT ZERO, DO AGAIN
021502  013746  002314                          MOV     DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
021506  112737  000377  002314                  MOVB    #377,DESTRK     ;SET LOW BYTE OF ADDRESS
021514  052737  100000  002236                  BIS     #CMOD,OFFSET    ;SET COMMAND MODIFIER BIT
021522  012737  031414  002234                  MOV     #TDWRD1,BUSADR  ;GET POINTER TO BEGINNING OF TRACK DESCRIPTOR BUFFER
021530  013737  002330  002376                  MOV     TDBCNT,NEGWRD   ;6 WORD TRANSFER
021536  004737  010502                          JSR     PC,DRIVER       ;NOW DO THE COMMAND
021542  012637  002314                          MOV     (SP)+,DESTRK    ;;POP STACK INTO DESTRK
021546  004737  013304                          JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
021552  000453                                  BR      7$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                        ;OTHERWISE RETURN HERE
021554  005737  002274                          TST     RTYCNT          ;DID WE HAVE AN ERROR ?
021560  001341                                  BNE     1$              ;BR IF YES
                                                                        ;READ WAS GOOD, SO NOW COMPARE EXPECTED
                                                                        ;AND RECIEVED DATA
021562  012700  031414                          MOV     #TDWRD1,R0      ;GET POINTER TO BEGINNING OF TD BUFFER
021566  013737  002312  031430                  MOV     DESCYL,TDCPY1   ;NOW PREPARE TO CHECK THE CORRECTNESS OF THE DATA
021574  052737  030000  031430                  BIS     #BIT13!BIT12,TDCPY1     ;THIS IS 16 BIT MODE, AND A TRACK DESCRIPTOR
021602  051037  031430                          BIS     (R0),TDCPY1     ;SET REMAINDER OF EXPECTED BITS
021606  013737  002314  031432                  MOV     DESTRK,TDCPY2   ;SET UP FOR THE SECOND WORD OF THE TD
021614  052737  000377  031432                  BIS     #377,TDCPY2     ;SET LOW BYTE OF ADDRESS
021622  022037  031430                          CMP     (R0)+,TDCPY1    ;CYLINDER ADDRESS MATCH ?
021626  001003                                  BNE     3$              ;BR IF NO
021630  022037  031432                          CMP     (R0)+,TDCPY2    ;TRACK/SECTOR ADDRESS MATCH ?
021634  001404                                  BEQ     4$              ;BR IF YES
021636  012737  000011  002272      3$:         MOV     #11,ERRMSK      ;LOG 'TD ADDRESS INCORRECT ERROR' STATUS
021644  000405                                  BR      5$              ;TRY READ AGAIN

021646  005720                      4$:         TST     (R0)+           ;IS IT VALID DATA ?
```

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

```
                                .SBTTL  LIST TD(S) MODULE

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

 021714                         LISTDS:
 021714  013746  002404                 MOV     FUNCTN,-(SP)    ;;PUSH FUNCTN ON STACK
 021720  010146                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 021722  010246                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 021724  010346                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 021726  023727  031420  140000         CMP     TDWRD3,#NULL    ;ARE THERE ANY DEFECTS TO LIST ?
 021734  001525                 BEQ     8$              ;BR IF NO
 021736  012737  006046  002360         MOV     #TDDEF,TEXT     ;LOAD TEXT WITH ADDRESS OF MESSAGE
 021744  005737  002204                 TST     LISHDR          ;LIST HEADERS IN DEFECT SECTORS ?
 021750  001403                 BEQ     1$              ;BR IF NO
 021752  012737  006103  002360         MOV     #ANDHDR,TEXT    ;LOAD TEXT WITH ADDRESS OF MESSAGE
 021760  004737  017240         1$:     JSR     PC,TDDUMP       ;GO DUMP THE TRACK DESCRIPTOR

 021764  005737  002204                 TST     LISHDR          ;LIST HEADERS IN DEFECT SECTORS ?
 021770  001507                 BEQ     8$              ;BR IF NO
 021772  012737  000073  002404         MOV     #RDHD,FUNCTN    ;LOAD A READ HEADER COMMAND
 022000  004737  015452                 JSR     PC,MAKTRK       ;NOW EXECUTE THE COMMAND
 022004  000501                 BR      8$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                        ;ELSE RETURN HERE
 022006  013701  002234                 MOV     BUSADR,R1       ;GET POINTER TO BEGINNING OF HEADER BLOCK
 022012  013703  002364                 MOV     TKWCNT,R3       ;GET WORD COUNT FOR TRACK

 022016  022121                 2$:     CMP     (R1)+,(R1)+     ;GET OVER HEADER WORDS 1 AND 2
 022020  021127  140000                 CMP     (R1),#NULL      ;ARE THERE ANY DEFECTS TO LIST IN THIS SECTOR ?
 022024  001003                 BNE     3$              ;BR IF YES
 022026  022121                 CMP     (R1)+,(R1)+     ;GET OVER HEADER WORDS 3 AND 4
 022030  022121                 CMP     (R1)+,(R1)+     ;GET OVER HEADER WORDS 5 AND 6
 022032  000463                 BR      7$
 022034  016146  177776         3$:     MOV     -2(R1),-(SP)    ;GET THE SECOND WORD OF THE TD (TRK/SEC ADRS)
 022040  011646                 MOV     (SP),-(SP)      ;SAVE SECOND WORD
 022042  105066  000003                 CLRB    3(SP)           ;GET RID OF HI BYTE (EXTRACT SECTOR)
                                                        ;PRINT TD WORD#2, SEC DATA
 022046  012646                 MOV     (SP)+,-(SP)
 022050  012646                 MOV     (SP)+,-(SP)
 022052  012746  004565                 MOV     #FRMT37,-(SP)
 022056  012746  000003                 MOV     #3,-(SP)
 022062  010600                 MOV     SP,R0
 022064  104417                 TRAP    C$PNTF
 022066  062706  000010                 ADD     #10,SP
 022072  012702  000004                 MOV     #4,R2           ;GET MAXIMUM # OF DEFECTS
 022076  022127  140000         4$:     CMP     (R1)+,#NULL     ;IS IT NULL DATA ?
 022102  001413                 BEQ     5$              ;BR IF YES
                                                        ;PRINT THE DEFECT DATA
```

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

```
                                .SBTTL  NEW DEFECT LIST DUMP MODULE
                                ;
                                ;;*****************************************************************************
                                ;THIS MODULE IS USED TO PRINT ANY NEW DEFECT DATA FOUND DURING A SCAN
                                ;OPERATION.
                                ;*
                                ;INPUTS ARE:    FROM A BUFFER POINTED BY R1
                                ;
                                ;OUTPUTS ARE.   PRESENTED TO THE USER ON THE CONSOLE PRINTER
                                ;*
                                ;CALL
                                ;       JSR     PC,NEWLST
                                ;;*****************************************************************************
                                ;
022224  010103                  NEWLST: MOV     R1,R3           ;GET THE BUFFER ADDRESS
022226  105237  002243                  INCB    SCANIT          ;INCREMENT SCAN ITERATION COUNT
022232  005046                          CLR      (SP)           ;SETUP SCAN ITERATION NUMBER FOR TYPEOUT
022234  113716  002243                  MOVB    SCANIT,(SP)     ;GET THE ITERATION COUNT
                                                                ;PRINT 'NEW DEFECT(S) DURING SCAN ITERATION XXX..
                                                                ;       TD(S) FOLLOW:'
022240  012646                          MOV     (SP)+,-(SP)
022242  012746  004253                  MOV     #FRMT32,-(SP)
022246  012746  000002                  MOV     #2,-(SP)
022252  010600                          MOV     SP,R0
022254  104417                          TRAP    C$PNTF
022256  062706  000006                  ADD     #6,SP
                                                                ;PRINT 'WORD#1 (CYL) WORD#2 (TRK)(SEC) WORD#3
                                                                ;       WORD#4 WORD#5 WORD#6'
022262  012746  004406                  MOV     #FRMT35,-(SP)
022266  012746  000001                  MOV     #1,-(SP)
022272  010600                          MOV     SP,R0
022274  104417                          TRAP    C$PNTF
022276  062706  000004                  ADD     #4,SP
022302  005713                  1$:     TST     (R3)            ;ANY MORE NEW DEFECTS ?
022304  001517                          BEQ     5$              ;BR IF NO
022306  062703  000014                  ADD     #<6*2>,R3       ;MOVE TO THE END OF THIS NEW DEFECT TD LIST
022312  032723  000400                  BIT     #BIT8,(R3)+     ;WAS DEFECT BIT SET IN STATUS ?
022316  001771                          BEQ     1$              ;BR IF NO
022320  005743                          TST     -(R3)           ;BACKUP R3 POINTER AND
022322  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
022324  012746  177777                  MOV     #-1,-(SP)       ;;PUSH #-1 ON STACK
022330  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022332  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022334  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022336  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022340  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022342  011646                          MOV     (SP),-(SP)      ;SAVE 2ND TD WORD
022344  105066  000003                  CLRB    3(SP)           ;GET RID OF HI BYTE (EXTRACT SECTOR)
022350  011646                          MOV     (SP),-(SP)      ;SAVE 2ND TD WORD
022352  105066  000002                  CLRB    2(SP)           ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
022356  000366  000002                  SWAB    2(SP)           ;MAKE IT THE LOW BYTE OF DATA.
022362  014346                          MOV     -(R3),-(SP)     ;;PUSH -(R3) ON STACK
022364  011646                          MOV     (SP),-(SP)      ;SAVE 1ST TD WORD
022366  042766  170000  000002          BIC     #170000,2(SP)   ;GET RID OF THE STATUS BITS
                                                                ;PRINT TD WORD#1, CYL, TD WORD#2 & TRK DATA
022374  012646                          MOV     (SP)+,-(SP)
022376  012646                          MOV     (SP)+,-(SP)
```

```
        022400  012646                          MOV     (SP)+,-(SP)
        022402  012646                          MOV     (SP)+, (SP)
        022404  012646                          MOV     (SP)+,-(SP)
        022406  012746  004515                  MOV     #FRMT36,-(SP)
        022412  012746  000006                  MOV     #6,-(SP)
        022416  010600                          MOV     SP,R0
        022420  104417                          TRAP    C$PNTF
        022422  062706  000016                  ADD     #16,SP
    42  022426  021627  140000          2$:     CMP     (SP),#NULL      ;IS IT NULL DATA ?
    43  022432  001412                          BEQ     3$              ;IF MATCH, YES
    44                                                                  ;PRINT THE DATA
    45  022434  012646                          MOV     (SP)+,-(SP)
        022436  012746  003640                  MOV     #FRMT14,-(SP)
        022442  012746  000002                  MOV     #2,-(SP)
        022446  010600                          MOV     SP,R0
        022450  104417                          TRAP    C$PNTF
        022452  062706  000006                  ADD     #6,SP
    46  022456  000411                          BR      4$              ;AND GO-ON
    47  022460                          3$:                             ;PRINT ' 140000 '
    48  022460  012646                          MOV     (SP)+,-(SP)
        022462  012746  003651                  MOV     #FRMT15,-(SP)
        022466  012746  000002                  MOV     #2,-(SP)
        022472  010600                          MOV     SP,R0
        022474  104417                          TRAP    C$PNTF
        022476  062706  000006                  ADD     #6,SP
    49  022502  021627  177777          4$:     CMP     (SP),#-1        ;IS THIS TERMINATOR ?
    50  022506  001347                          BNE     2$              ;BR IF NO
    51  022510  005726                          TST     (SP)+           ;RESTORE STACK
    52                                                                  ;CR-LF
    53  022512  012746  002656                  MOV     #CRLF,-(SP)
        022516  012746  000001                  MOV     #1,-(SP)
        022522  010600                          MOV     SP,R0
        022524  104417                          TRAP    C$PNTF
        022526  062706  000004                  ADD     #4,SP
    54  022532  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
    55  022534  005723                          TST     (R3)+           ;GET OVER INTERNAL STATUS WORD IN TD LIST
    56  022536  020327  120072                  CMP     R3,#ENDBUF      ;AT END OF BUFFER?
    57  022542  101657                          BLOS    1$              ;BR IF NO
    58  022544  000207                  5$:     RTS     PC
```

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

```
                                .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
                                ;               A BUFFER CALLED 'MFGBUF'.
                                ;*
                                ;CALL
                                ;       JSR     PC,HDSCAN
                                ;;*****************************************************************************
                                ;
022732  012737  000012  002264  HDSCAN: MOV     #10.,ITCNT      ;INITIALIZE ITERATION COUNT
022740  013702  002270                  MOV     ERRTTL,R2       ;SAVE ERROR TOTAL AT THIS POINT
022744  005003                        CLR     R3              ;CLEAR HEADER COUNTER
022746  012704  033614                MOV     #TDSBLK,R4      ;GET POINTER TO BEGINNING OF DEFECT BUFFER
022752  012737  000073  002404  1$:    MOV     #RDHD,FUNCTN    ;SETUP FOR READ-HEADER FUNCTION
022760  012737  005713  002260        MOV     #RHDATA,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
022766  012737  036074  002234        MOV     #DBUFF,BUSADR   ;GET POINTER TO BEGINNING OF DATA BUFFER
022774  013737  002364  002376        MOV     TKWCNT,NEGWRD   ;SET TRACK WORD COUNT (6 WRDS X 50. SECTORS)
023002  052737  100000  002236        BIS     #CMOD,OFFSET    ;SET COMMAND MODIFIER BIT
023010  004737  010502                JSR     PC,DRIVER       ;DO THE TRANSFER
023014  004737  013304                JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
023020  000525                        BR      13$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                            ;ELSE RETURN HERE
023022  005737  002274                TST     RTYCNT          ;DID WE HAVE AN ERROR ?
023026  001405                        BEQ     2$              ;BR IF NO
023030  032737  020600  002530        BIT     #OPI!HCE!HCRC,REG+14    ;IS IT AN OPI, HCE OR HCRC ERROR ?
023036  001012                        BNE     5$              ;BR IF YES
023040  000403                        BR      3$
                                ;
023042  023737  002270        2$:     CMP     ERRTTL,R2       ;DID WE HAVE AN ERROR BEFORE ?
023046  001403                        BEQ     4$              ;BR IF NO
023050  005737  002264        3$:     TST     ITCNT           ;DONE WITH ITERATIONS YET ?
023054  003336                        BGT     1$              ;BR IF NO
023056  062716  000002        4$:     ADD     #2,(SP)         ;RETURN WITH NO ERROR (HEADER OK).
023062  000515                        BR      16$
                                ;
023064  005203                5$:     INC     R3              ;INCREMENT HEADER COUNT
023066  005037  002274                CLR     RTYCNT          ;RESET RETRY COUNT TO 0
023072  004737  024116                JSR     PC,FINDSEC      ;GO FIND THE SECTOR IN WHICH THE NEW DEFECT
                                                            ;WAS FOUND.
023076  001020                        BNE     7$              ;BR IF NOT SECTOR 0, ELSE
                                                            ;SEE WHAT CONSTANT HAS TO BE USED TO MOVE
                                                            ;HEADER 0
023100  032737  100000  031414        BIT     #BIT15,TDWRD1   ;IS HEADER 0 ALREADY MOVED ?
023106  001004                        BNE     6$              ;BR IF YES
023110  012737  000243  002400        MOV     #<G1+TD+G2+<DS/2>>,RELWRD       ;GET THE CONSTANT FOR HDR 0 MOVED
023116  000437                        BR      11$             ;INSERT DEFECT INTO TD
                                ;
023120  032737  040000  031414  6$:    BIT     #BIT14,TDWRD1   ;IS THE TD ALREADY MOVED ?
023126  001054                        BNE     12$             ;BR IF YES
023130  012737  000166  002400        MOV     #<G1+<DS/2>>,RELWRD     ;GET THE CONSTANT FOR TD MOVED
023136  000427                        BR      11$             ;INSERT DEFECT INTO TD
                                                            ;NOW BACKUP THE SECTOR ADDRESS IN INTERLEAVED
```

```
                                                        ;OR NON INTERLEAVED MODE.
023140  113705  002323          7$:     MOVB    PLTRK,R5        ;GET 1ST INTERLEAVED SECTOR
023144  005305                  DEC     R5              ;BACKUP ONE SECTOR
023146  105737  002244          TSTB    INTLEV          ;INTERLEAVED SECTORS ?
023152  001414                  BEQ     9$              ;BR IF NO
023154  032737  020000  002530  BIT     #OPI,REG+14     ;IS IT AN OPI ERROR ?
023162  001010                  BNE     9$              ;BR IF YES
023164  123705  002402          CMPB    SECADD,R5       ;IS THIS SECTOR > 24. ?
023170  003003                  BGT     8$              ;BR IF YES
023172  060537  002402          ADD     R5,SECADD       ;ADJUST SECTOR ADRS FOR A SECTOR <= 24.
023176  000404                  BR      10$
023200  160537  002402          8$:     SUB     R5,SECADD       ;ADJUST SECTOR ADRS FOR A SECTOR  > 24.
023204  005337  002402          9$:     DEC     SECADD                  ;BACKUP THE SECTOR COUNT TO
023210  012737  000470  002400  10$:    MOV     #<DATA+<DS/2>>,RELWRD   ;PUT THE SKIP IN GAP 4 OF THE
                                                                        ;SECTOR PRECEEDING THE SECTOR
                                                                        ;WITH THE BAD HEADER.
023216  004737  016536          11$:    JSR     PC,RELTIVE      ;CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
023222  013714  002400          MOV     RELWRD,(R4)     ;LOG THE NEW DEFECT IN DEFECT BUFFER
023226  004737  024150          JSR     PC,TOLER        ;SEE IF HEADER ERROR'S ARE WITHIN TOLERANCE
                                                        ;AND UPDATE POINTER TO DEFECT TABLE IN R4.
023232  005737  002272          TST     ERRMSK          ;DID RANDOM HEADER ERRORS OCCUR ?
023236  100422                  BMI     14$             ;BR IF YES
023240  020327  000002          CMP     R3,#2           ;DID WE HAVE 2 MORE HEADER ERRORS ?
023244  003642                  BLE     1$              ;BR IF NO
023246  004737  016616          JSR     PC,INSERT       ;INSERT DEFECT INTO TD

023252  005737  031460          TST     TMPBSF          ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
023256  100423                  BMI     17$             ;BR IF YES
023260  113737  002402  031462  12$:    MOVB    SECADD,TMPBSF+2 ;SAVE THE BAD SECTOR AND
023266  004737  011476          JSR     PC,LODBSF       ;LOAD THE BSF BUFFER
023272  000415                  BR      17$

023274  012737  000017  002272  13$:    MOV     #17,ERRMSK      ;LOG 'UNRECOVERABLE ERROR DURING PACK SCAN' STATUS
023302  000403                  BR      15$             ;TAKE THE BAD RETURN
023304  012737  000016  002272  14$:    MOV     #16,ERRMSK      ;LOG 'RANDOM HEADER ERROR' STATUS
023312  005037  002252          15$:    CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
023316  042761  000400  000014  16$:    BIC     #BIT8,14(R1)    ;CLEAR INTERNAL DEFECT STATUS AND
023324  000207                  RTS     PC              ;TAKE HARD ERROR OR NO ERROR RETURN.

023326  105237  002246          17$:    INCB    NEWCNT          ;INCREMENT THE NEW DEFECT COUNT
023332  012704  000006          MOV     #6,R4           ;GET # OF ENTRIES IN TRACK DESCRIPTOR
023336  012703  031414          MOV     #TDWRD1,R3      ;GET POINTER TO BEGINNING OF TD BUFFER
023342  012321                  18$:    MOV     (R3)+,(R1)+     ;PUT NEW DEFECT INTO BUFFER
023344  005304                  DEC     R4              ;DONE ALL ENTRIES YET ?
023346  003375                  BGT     18$             ;BR IF NO
023350  032737  100000  031414  BIT     #BIT15,TDWRD1   ;IS HDR 0 MOVED ?
023356  001402                  BEQ     19$             ;BR IF NO
023360  052711  100000          BIS     #BIT15,(R1)     ;MARK HDR 0 MOVED, IN THE INTERNAL STATUS
023364  032737  040000  031414  19$:    BIT     #BIT14,TDWRD1   ;IS THE TD MOVED ?
023372  001402                  BEQ     20$             ;BR IF NO
023374  052711  040000          BIS     #BIT14,(R1)     ;MARK TD MOVED AND
023400  052721  000400          20$:    BIS     #BIT8,(R1)+     ;MARK DEFECT, IN THE INTERNAL STATUS.
023404  000207                  RTS     PC
```

```
                                .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
                                ;               A BUFFER CALLED 'MFGBUF'.
                                ;*
                                ;CALL
                                ;       JSR     PC,DASCAN
                                ;;*****************************************************************************

023406  012737  002142  002326  DASCAN: MOV     #BGNPAT,PATTRN  ;START WITH BEGINNING DATA PATTERN
023414  004737  024262          1$:     JSR     PC,GETPAT       ;GET A PATTERN AND SETUP 'PATTRN' TO CONTAIN
                                                                ;THE NEXT DATA PATTERN TO BE USED.
023420  012704  033614          2$:     MOV     #TDSBLK,R4      ;GET POINTER TO BEGINNING OF DEFECT BUFFER
023424  012737  000061  002404          MOV     #WRTDAT,FUNCTN  ;SETUP FOR A WRITE DATA COMMAND
023432  012737  005625  002260          MOV     #WRDAT,ERRFNC   ;SAVE FUNCTION FOR ERROR ROUTINE
023440  012737  036074  002234          MOV     #DBUFF,BUSADR   ;GET POINTER TO BEGINNING OF DATA BUFFER
023446  012737  031000  002376          MOV     #<256.*50.>,NEGWRD  ;SET FOR A FULL TRACK DATA TRANSFER
023454  004737  010502                  JSR     PC,DRIVER       ;DO THE TRANSFER
023460  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
023464  000520                          BR      10$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
023466  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
023472  001352                          BNE     2$              ;BR IF YES
                                                                ;GOOD WRITE!!!

                                        ;NOW DO A WRITE CHECK ON THE TRACK JUST WRITTEN

023474  012737  000012  002264  3$:     MOV     #10.,ITCNT      ;INITIALIZE ITERATION COUNT
023502  013702  002270                  MOV     ERRTTL,R2       ;SAVE ERROR TOTAL AT THIS POINT
023506  005003                          CLR     R3              ;CLEAR WCE COUNTER
023510  012737  000051  002404  4$:     MOV     #WCKD,FUNCTN    ;SET UP FOR A WRITE-CHECK COMMAND
023516  012737  005636  002260          MOV     #WCKDAT,ERRFNC  ;SAVE FUNCTION FOR ERROR ROUTINE
023524  012737  031000  002376          MOV     #<256.*50.>,NEGWRD  ;FULL TRACK DATA TRANSFER
023532  004737  010502                  JSR     PC,DRIVER       ;DO THE WRITE CHECK NOW!
023536  004737  013304                  JSR     PC,RETRY        ;SEE IF WE HAD ANY ERRORS
023542  000471                          BR      10$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE
023544  005737  002274                  TST     RTYCNT          ;DID WE HAVE AN ERROR ?
023550  001405                          BEQ     5$              ;BR IF NO
023552  032737  040000  002524          BIT     #WCE,REG+10     ;WAS IT A WRITE CHECK ERROR ?
023560  001016                          BNE     8$              ;BR IF YES
023562  000403                          BR      6$              ;BR IF NO

023564  023702  002270          5$:     CMP     ERRTTL,R2       ;DID WE HAVE AN ERROR BEFORE ?
023570  001403                          BEQ     7$              ;BR IF NO
023572  005737  002264          6$:     TST     ITCNT           ;DONE WITH ITERATIONS YET ?
023576  003344                          BGT     4$              ;BR IF NO
023600  023727  002326  002156  7$:     CMP     PATTRN,#ENDPAT  ;DONE WITH ALL PATTERNS YET ?
023606  103702                          BLO     1$              ;BR IF NO
023610  062716  000002                  ADD     #2,(SP)         ;RETURN WITH NO ERROR (DATA OK).
023614  000455                          BR      13$
```

```
                                                ;NOW PROCESS THE WRITE CHECK ERROR, TO FIND BAD WORD

023616  005203                  8$:     INC     R3              ;INCREMENT WCE COUNT
023620  005037  002274                  CLR     RTYCNT          ;RESET RETRY COUNT TO 0
023624  004737  024040                  JSR     PC,FINDWRD      ;GO FIND THE BAD WORD WITHIN THE SECTOR OF THE
                                                                ;NEW DEFECT.
023630  004737  024116                  JSR     PC,FINDSEC      ;GO FIND THE SECTOR IN WHICH THE NEW DEFECT
                                                                ;WAS FOUND.
023634  004737  016536                  JSR     PC,RELTIVE      ;CALCULATE THE DEFECT VALUE RELATIVE TO INDEX
023640  013714  002400                  MOV     RELWRD,(R4)     ;LOG THE NEW DEFECT IN DEFECT BUFFER
023644  004737  024150                  JSR     PC,TOLER        ;SEE IF WCE'S ARE WITHIN TOLERANCE AND UPDATE
                                                                ;POINTER TO DEFECT TABLE IN R4.
023650  005737  002272                  TST     ERRMSK          ;DID RANDOM WCE'S OCCUR ?
023654  100430                          BMI     11$             ;BR IF YES
023656  020327  000002                  CMP     R3,#2           ;DID WE HAVE 2 WCE'S AFTER THE ORIGINAL ?
023662  003712                          BLE     4$              ;BR IF NO
023664  023737  002400  002354          CMP     RELWRD,TDLMTS+2 ;WILL BEGINNING OF DEFECT BE IN GAP 2 ?
023672  003005                          BGT     9$              ;BR IF NO
023674  013737  002354  002400          MOV     TDLMTS+2,RELWRD ;GET UPPER LIMIT FOR HDR 0 MOVED AND
023702  005237  002400                  INC     RELWRD          ;ADD ONE TO IT.
023706  004737  016616          9$:     JSR     PC,INSERT       ;INSERT DEFECT INTO TD

023712  005737  031460                  TST     TMPBSF          ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
023716  100420                          BMI     14$             ;BR IF YES
023720  004737  011476                  JSR     PC,LODBSF       ;LOAD THE BSF BUFFER
023724  000415                          BR      14$             ;TAKE THE RETURN

023726  012737  000017  002272  10$:     MOV     #17,ERRMSK      ;LOG 'UNRECOVERABLE ERROR DURING PACK SCAN' STATUS
023734  000403                          BR      12$             ;TAKE THE BAD RETURN
023736  012737  000020  002272  11$:     MOV     #20,ERRMSK      ;LOG 'RANDOM WCE ERROR' STATUS
023744  005037  002252          12$:     CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
023750  042761  000400  000014  13$:     BIC     #BIT8,14(R1)    ;CLEAR INTERNAL DEFECT STATUS AND
023756  000207                          RTS     PC              ;TAKE HARD ERROR OR NO ERROR RETURN.

023760  105237  002246          14$:     INCB    NEWCNT          ;INCREMENT THE NEW DEFECT COUNT
023764  012704  000006                  MOV     #6,R4           ;GET # OF ENTRIES IN TRACK DESCRIPTOR
023770  012703  031414                  MOV     #TDWRD1,R3      ;GET POINTER TO BEGINNING OF TD BUFFER
023774  012321                  15$:     MOV     (R3)+,(R1)+     ;PUT THE NEW DEFECT INTO BUFFER
023776  005304                          DEC     R4              ;DONE ALL ENTRIES YET ?
024000  003375                          BGT     15$             ;BR IF NO
024002  032737  100000  031414          BIT     #BIT15,TDWRD1   ;IS HDR 0 MOVED ?
024010  001402                          BEQ     16$             ;BR IF NO
024012  052711  100000                  BIS     #BIT15,(R1)     ;MARK HDR 0 MOVED, IN THE INTERNAL STATUS
024016  032737  040000  031414  16$:     BIT     #BIT14,TDWRD1   ;IS THE TD MOVED ?
024024  001402                          BEQ     17$             ;BR IF NO
024026  052711  040000                  BIS     #BIT14,(R1)     ;MARK TD MOVED AND
024032  052721  000400          17$:     BIS     #BIT8,(R1)+     ;MARK DEFECT, IN THE INTERNAL STATUS.
024036  000207                          RTS     PC              ;TAKE RETURN
```

```
                                ;;*****************************************************************************
                                ;THIS MODULE IS USED BY 'DASCAN' TO IDENTIFY AND LOCATE ANY DEFECT INFORMATION
                                ;FOUND DURING A SCAN OPERATION.  THE MODULE DETERMINES THE SECTOR ADDRESS OF
                                ;THE NEW DEFECT AND THE WORD COUNT WITHIN THE SECTOR OF THE NEW DEFECT.
                                ;*
                                ;INPUTS ARE:    REG+2 (RPWC)
                                ;
                                ;OUTPUTS ARE:   RELWRD (WORD COUNT IN SECTOR)
                                ;*
                                ;CALL
                                ;       JSR     PC,FINDWRD      ;CALL THE ROUTINE
                                ;;*****************************************************************************

024040  013746  002516          FINDWRD: MOV    REG+2,-(SP)     ;PUT RPWC CONTENTS ON THE STACK
024044  062716  031000                   ADD    #<256.*50.>,(SP)        ;AND CALCULATE THE # OF WORDS XFERD
024050  005316                           DEC    (SP)            ;SUBTRACT 1 FOR AN ODD OR EVEN WORD
024052  005737  002432                   TST    RHTYPE          ;WAS THE CONTROLLER AN RH70 ?
024056  001407                           BEQ    1$              ;BR IF 0, NO
024060  162716  000002                   SUB    #2,(SP)         ;SUBTRACT 2 FOR A DOUBLE WORD
024064  032737  004000  002566           BIT    #BIT11,REG+52   ;WAS WCE ON AN EVEN WORD?
024072  001401                           BEQ    1$              ;BR IF 0, NO
024074  005316                           DEC    (SP)            ;SUBTRACT 1 FOR AN EVEN WORD
024076  162716  000400          1$:      SUB    #256.,(SP)      ;DID WE FIND THE BAD SECTOR YET ?
024102  002375                           BGE    1$              ;BR IF NO
024104  062716  000400                   ADD    #256.,(SP)      ;THIS IS THE BAD WORD IN THE BAD SECTOR
024110  012637  002400                   MOV    (SP)+,RELWRD    ;SAVE WORD COUNT IN SECTOR
024114  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
                                ;;*****************************************************************************

024116  005037  002402          FINDSEC: CLR    SECADD          ;INIT SECTOR ADDRESS COUNT
024122  113737  002522  002402           MOVB   REG+6,SECADD    ;LOAD SECTOR ADDRESS IN ERROR
024130  001004                           BNE    1$              ;BR IF ERROR NOT IN LAST SECTOR, ELSE
024132  113737  002322  002402           MOVB   SEC50,SECADD    ;SETUP FOR ERROR IN LAST SECTOR
024140  000402                           BR     2$

024142  005337  002402          1$:      DEC    SECADD          ;ADJUST RPDA TO ACCOUNT FOR ERROR
024146  000207                  2$:      RTS    PC              ;AND TAKE THE RETURN
```

```
                                ;;*****************************************************************************
                                ;THIS MODULE IS USED TO DETERMINE IF THE DATA FOUND DURING A SCAN OPERATION
                                ;IS DATA WHICH OCCURS WITH CONSISTENT RESULTS.  IF THE DATA BUFFER CONTAINS
                                ;DATA FOR ANY ONE TRACK WHICH DEVIATES MORE THAN +/- 3 WORDS, THE FAILURE IS
                                ;CONSIDERED TO BE A RANDOM WRITE CHECK ERROR, THE DEFECT IS NOT LOGGED IN THE
                                ;TRACK DESCRIPTOR AND THE USER IS NOTIFIED OF THE PROBLEM.
                                ;*
                                ;INPUTS ARE:    FROM A BUFFER CALLED 'TDSBLK'
                                ;
                                ;OUTPUTS ARE:   RELWRD (AVERAGE DEFECT), ERRMSK (IF ERROR DETECTED)
                                ;*
                                ;CALL
                                ;       JSR     PC,TOLER        ;CALL ROUTINE
                                ;;*****************************************************************************

024150  012700  024256        TOLER:  MOV     #5$,R0          ;GET POINTER TO HI VALUE FOR WCE'S
024154  022704  033614                CMP     #TDSBLK,R4      ;IS THIS THE FIRST WCE DETECTED ?
024160  001003                        BNE     1$              ;BR IF NO
024162  011410                        MOV     (R4),(R0)       ;SAVE THIS DEFECT AS HI AND
024164  011460  000002                MOV     (R4),2(R0)      ;LO VALUES.
024170  021410                1$:     CMP     (R4),(R0)       ;IS THIS VALUE HIGHER THAN PREVIOUS HI ?
024172  002402                        BLT     2$              ;BR IF NO
024174  011410                        MOV     (R4),(R0)       ;SAVE THIS AS HI VALUE NOW
024176  000405                        BR      3$
024200  005720                2$:     TST     (R0)+           ;SET POINTER TO LO VALUE
024202  021410                        CMP     (R4),(R0)       ;IS THIS VALUE LOWER THAN PREVIOUS LO ?
024204  003002                        BGT     3$              ;BR IF NO
024206  011410                        MOV     (R4),(R0)       ;SAVE THIS AS LO VALUE NOW
024210  005740                        TST     -(R0)           ;SET POINTER TO HI VALUE
024212  012046                3$:     MOV     (R0)+,-(SP)     ;GET HI VALUE AND
024214  011646                        MOV     (SP),-(SP)      ;SAVE IT.
024216  061016                        ADD     (R0),(SP)       ;TOTAL THE HI AND LO VALUES AND
024220  000241                        CLC                     ;CLEAR CARRY BIT
024222  006016                        ROR     (SP)            ;DETERMINE THE AVERAGE.
024224  012637  002400                MOV     (SP)+,RELWRD    ;SAVE AVERAGE VALUE AS DEFECTIVE SPOT
024230  161016                        SUB     (R0),(SP)       ;GET DIFFERENCE FROM LO TO HI VALUES
024232  022627  000006                CMP     (SP)+,#6        ;ARE HI/LO VALUES WITHIN -+3 WORDS ?
024236  003405                        BLE     4$              ;BR IF YES
024240  005037  002252                CLR     SUPRSS          ;ALLOW ERROR MESSAGE TO BE OUTPUT
024244  012737  177777  002272        MOV     #-1,ERRMSK      ;LOG 'RANDOM ERROR' STATUS
024252  005724                4$:     TST     (R4)+           ;UPDATE WCE TABLE POINTER
024254  000207                        RTS     PC              ;AND RETURN

024256  000000                5$:     .WORD   0               ;HI VALUE GOES HERE
024260  000000                6$:     .WORD   0               ;LO VALUE GOES HERE
```

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

```
                                .SBTTL  SCRATCH ROUINES

                                ;;*****************************************************************************
                                ;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.
                                ;*
                                ;CALL
                                ;       JSR     PC,TOLRAN
                                ;;*****************************************************************************

024336                          TOLRAN:
024336  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024340  013746  002366                  MOV     TEMP1,-(SP)     ;;PUSH TEMP1 ON STACK
024344  013746  002370                  MOV     TEMP2,-(SP)     ;;PUSH TEMP2 ON STACK
024350  012701  000003                  MOV     #3,R1           ;THIS IS THE TOLERANCE VALUE
024354  023737  002366  002370  1$:     CMP     TEMP1,TEMP2     ;DO THE INPUTS MATCH?
024362  001415                          BEQ     4$              ;IF EQUAL YES -RESTORE STACK AND RETURN
024364  103403                          BLO     2$              ;THEY DO NOT, TEMP 1< TEMP2 (BRANCH)
024366  005237  002370                  INC     TEMP2           ;ADD ONE TO TEMP2 (TOO SMALL)
024372  000402                          BR      3$              ;AND REDUCE THE ITERATIONS LEFT
024374  005337  002370          2$:     DEC     TEMP2           ;SUBTRACT ONE FROM TEMP 2 (TOO LARGE)
024400  005301                  3$:     DEC     R1              ;ONE LESS ITERATION TO GO
024402  003364                          BGT     1$              ;OF >0 DO AGAIN
024404  012637  002370                  MOV     (SP)+,TEMP2     ;;POP STACK INTO TEMP2
024410  012637  002366                  MOV     (SP)+,TEMP1     ;;POP STACK INTO TEMP1
024414  000401                          BR      5$              ;AND RETURN
024416  022626                  4$:     CMP     (SP)+,(SP)+     ;POP STACK TWICE
024420                          5$:
024420  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
024422  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 EXIST.
                                ;*
                                ;CALL
                                ;       JSR     PC,CHEKTD
                                ;;*****************************************************************************

024424                          CHEKTD:
024424  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024426  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024430  062701  000004                  ADD     #4,R1           ;MOVE R1 UP TO THE FIRST DEFECT
024434  005037  002370                  CLR     TEMP2           ;TEMP2 IS THE DEFECT DATA ACCUMULATOR
024440  012702  000004                  MOV     #4,R2           ;ALLOW 4 DEFECTS MAX!
024444  062137  002370          1$:     ADD     (R1)+,TEMP2     ;ACCUMULATE A DEFECT
024450  004737  024336                  JSR     PC,TOLRAN       ;CHECK ITS TOLERANCE
024454  005737  002370                  TST     TEMP2           ;IF IT'S OK, TEMP2 <>0!
024460  001002                          BNE     2$              ;IF OK, TAKE BRANCH
024462  005302                          DEC     R2              ;ONE LESS DATA ITEM
024464  003367                          BGT     1$              ;IF >0, KEEP GOING
024466                          2$:
024466  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
024470  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
024472  000207                          RTS     PC              ;RETURN
```

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

024474                          FINDBAD:
024474  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024476  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024500  012701  031420                  MOV     #TDWRD3,R1      ;GET THE FIRST DEFECT
024504  012702  000004                  MOV     #4,R2           ;AND THE RECORD SIZE
024510  062137  002400          1$:     ADD     (R1)+,RELWRD    ;ADD THE LIST OF DEFECT TO THE NEW DEFECT
024514  005302                          DEC     R2              ;ONE LESS RECORD ENTRY
024516  003374                          BGT     1$              ;OF >0, KEEP GOING
024520  012701  000155                  MOV     #<G1+TD+G2>,R1  ;THE # OF WORDS BEFORE HDR 0
024524  062701  000517                  ADD     #<HDR+G3+DATA+G4>,R1    ;ADD THE OVERHEAD FOR A SECTOR
024530  005037  002402                  CLR     SECADD          ;DEFAULT=SECTOR 0
024534  020137  002400                  CMP     R1,RELWRD       ;NOW SCALE THE VALUES
024540  103006                          BHIS    3$              ;IF R1 IS GREATER, SECTOR 0 IS BAD
024542  005237  002402          2$:     INC     SECADD          ;NEXT SECTOR ADDRESS
024546  162737  000517  002400        SUB     #<HDR+G3+DATA+G4>,RELWRD        ;REDUCE THE REMAINING WORD COUNT
024554  100372                          BPL     2$              ;OF <0, WE'VE FOUND THE BAD  SECTOR
024556                          3$:
024556  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
024560  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
024562  000207                          RTS     PC              ;RETURN TO CALLER
```

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

024702                          GAPVAL:
024702  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024704  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024706                          1$:
024706  011146                        MOV     (R1),-(SP)      ;;PUSH (R1) ON STACK
024710  042716  170000                BIC     #170000,(SP)    ;STRIP THE UNWANTED BITS
024714  022637  002372                CMP     (SP)+,HICYL     ;MATCH?
024720  001406                        BEQ     2$              ;IF EQUAL, THEY MATCH
024722  062701  000016                ADD     #<7*2>,R1       ;GET NEXT BUFFER ENTRY
024726  020127  036072                CMP     R1,#ENDDEF      ;OUT OF DEFECT BUFFER YET ?
024732  101765                        BLOS    1$              ;BR IF NO
024734  000453                        BR      6$              ;DONE, IF OVERRUN

024736  126137  000003  002315  2$:   CMPB    3(R1),DESTRK+1  ;IS THIS THE CORRECT TRACK ADDRESS?
024744  001406                        BEQ     3$              ;IF =, YES
024746  062701  000016                ADD     #<7*2>,R1       ;GET NEXT BUFFER ENTRY
024752  020127  036072                CMP     R1,#ENDDEF      ;OUT OF DEFECT BUFFER YET ?
024756  101767                        BLOS    2$              ;BR IF NO
024760  101041                        BHI     6$              ;DONE IF OVERRUN
024762  012702  000005          3$:   MOV     #5,R2           ;GET THE DELTA VALUE
024766  011137  002312                MOV     (R1),DESCYL     ;SAVE THE CURRENT CYLINDER ADDRESS
024772  005237  002312                INC     DESCYL          ;AND LOOK FOR THE NEXT VALUE
024776  062701  000016          4$:   ADD     #<7*2>,R1       ;GET NEXT BUFFER VALUE
025002  020127  036072                CMP     R1,#ENDDEF      ;OUT OF DEFECT BUFFER YET ?
025006  101026                        BHI     6$              ;BR IF YES
025010  021137  002312                CMP     (R1),DESCYL     ;CYLINDER ADDRESS MATCH?
025014  103770                        BLO     4$              ;TAKE BRANCH IF NOT
025016  126137  000003  002315        CMPB    3(R1),DESTRK+1  ;TRACK ADDRESS MATCH?
025024  001405                        BEQ     5$              ;IF SO TAKE BRANCH
025026  005237  002312                INC     DESCYL          ;NEXT CYL ADDRESS-PLEASE
025032  005302                        DEC     R2              ;ONE LESS ITERATION TO GO
025034  003360                        BGT     4$              ;IF >0, WE'RE NOT DONE
025036  000412                        BR      6$              ;IF 0 WE NEVER FOUND A CONTINUATION
025040  004737  024424          5$:   JSR     PC,CHEKTD       ;GET THE NEW TD SUMMATION
025044  005737  002370                TST     TEMP2           ;IF A SCRATCH VALUE FOUND, THIS <>0
025050  001405                        BEQ     6$              ;TAKE BRANCH IF SCRATCH VALUE NOT FOUND
025052  012746  000005                MOV     #5,-(SP)        ;GET THE MAX VALUE FOR INTERPOLATION
025056  160216                        SUB     R2,(SP)         ;AND DERIVE THE IMPLIED SCRATCH WIDTH
025060  062637  002372                ADD     (SP)+,HICYL     ;ADD THAT WIDTH TO THE HIGH LIMIT
025064                          6$:
025064  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
025066  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
025070  000207                        RTS     PC              ;RETURN TO CALLER
```

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

025072                          HIVALU:
025072  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025074  062701  000016                  ADD     #<7*2>,R1       ;MOVE INPUT OVER THE FIRST DEFECT
025100  126137  000003  002315  1$:     CMPB    3(R1),DESTRK+1  ;TRACK MATCH ?
025106  001007                          BNE     2$              ;IF NOT, TAKE BRANCH
025110  004737  024424                  JSR     PC,CHEKTD       ;DOES THE TD DATA MATCH?
025114  005737  002370                  TST     TEMP2           ;IF TEMP2=0,NO!
025120  001402                          BEQ     2$              ;TAKE BRANCH IF NO MATCH
025122  005237  002372                  INC     HICYL           ;INCLUDE THIS CYLINDER IN THE SCRATCH AREA
025126  062701  000016          2$:     ADD     #<7*2>,R1       ;NEXT ENTRY
025132  020127  036072                  CMP     R1,#ENDDEF      ;OUT OF DEFECT BUFFER YET ?
025136  101760                          BLOS    1$              ;BR IF NO
025140  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025142  000207                          RTS     PC              ;RETURN

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

025144                          COMPRS:
025144  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025146  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
025150  010102                          MOV     R1,R2           ;GET THE CURRENT BUFFER ADDRESS
025152  062702  000016                  ADD     #<7*2>,R2       ;GET THE NEXT ENTRY
025156  012221          1$:     MOV     (R2)+,(R1)+     ;COMPRESS OVER THE CURRENT ENTRY
025160  020227  036072                  CMP     R2,#ENDDEF      ;OUT OF DEFECT BUFFER YET ?
025164  101774                          BLOS    1$              ;BR IF NO
025166  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
025170  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025172  000207                          RTS     PC
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE OUTPUTS DATA TO THE NEW DEFECT BUFFER FOR EVENTUAL TD REWRITE OR
 4                                ;IT PRODUCES OUTPUT TO THE BAD SECTOR FILE WHEN A TD IS FOUND TO CONTAIN 4
 5                                ;DEFECTS.
 6                                ;*
 7                                ;OUTPUTS ARE:   TO (R2)+, A BUFFER CALLED 'MFGBUF' OR TO (R4)+, A BUFFER
 8                                ;               CALLED 'DBUFF'
 9                                ;*
10                                ;CALL
11                                ;       JSR     PC,UPDSCR
12                                ;;*****************************************************************************
13
14 025174                        UPDSCR:
   025174  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   025176  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
15 025200  004737  021440                JSR     PC,READTD       ;CALL THE READ TD SUBROUTINE
16 025204  000401                        BR      1$              ;RETURN HERE IF EXCEEDED RETRY LIMIT
17 025206  000402                        BR      2$              ;ELSE RETURN HERE, NO ERROR
18 025210  004737  026332        1$:     JSR     PC,DESTD        ;GET DESIRED TD, BECAUSE WE FAILED THE READ
19 025214                        2$:
   025214  013746  002400                MOV     RELWRD,-(SP)    ;;PUSH RELWRD ON STACK
20 025220  004737  016616                JSR     PC,INSERT       ;INSERT THE DEFECT IN TD
21 025224  012637  002400                MOV     (SP)+,RELWRD    ;;POP STACK INTO RELWRD
22 025230  005737  031460                TST     TMPBSF          ;DO WE HAVE ROOM IN THE TRACK DESCRIPTOR ?
23 025234  100410                        BMI     3$              ;BR IF YES
24 025236  004737  024474                JSR     PC,FINDBAD      ;FIND THE BAD SECTOR
25 025242  113737  002402  031462        MOVB    SECADD,TMPBSF+2 ;SAVE THE BAD SECTOR AND
26 025250  004737  011476                JSR     PC,LODBSF       ;LOAD THE BSF BUFFER
27 025254  000413                        BR      5$
28
29 025256  105237  002246        3$:     INCB    NEWCNT          ;ADD ONE TO THE TOTAL DEFECT COUNT
30 025262  012703  000006                MOV     #6,R3           ;GET # OF ENTRIES IN TRACK DESCRIPTOR
31 025266  012701  031414                MOV     #TDWRD1,R1      ;GET POINTER TO BEGINNING OF TD BUFFER
32 025272  012124                4$:     MOV     (R1)+,(R4)+     ;PUT THE NEW DEFECT INTO BUFFER
33 025274  005303                        DEC     R3              ;DONE ALL ENTRIES YET ?
34 025276  003375                        BGT     4$              ;BR IF NO
35 025300  052724  001000                BIS     #BIT9,(R4)+     ;MARK A SCRATCH, IN THE INTERNAL STATUS
36 025304                        5$:
   025304  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
   025306  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
37 025310  000207                        RTS     PC              ;NOW RETURN
```

```
 1
 2                                ;;*****************************************************************************
 3                                ;THIS MODULE IDENTIFIES DATA WHICH REPRESENTS A SCRATCH ON THE MEDIA.  THE
 4                                ;DATA FOUND MUST INITIALLY BE CONTIGUOUS, ON THE SAME TRACK - ADJACENT
 5                                ;CYLINDERS.  GAPS AND THE MAXIMUM AND MINIMUM INTERPOLATED ADDRESSES ARE THEN
 6                                ;DERIVED AND LOGGED IN  THE NEW DEFECT BUFFER FOR EVENTUAL TO REWRITE.
 7                                ;*
 8                                ;CALL
 9                                ;       JSR     PC,FINSCR
10                                ;;*****************************************************************************
11
12 025312                        FINSCR:
   025312  013746  002312                MOV     DESCYL,-(SP)    ;;PUSH DESCYL ON STACK
   025316  013746  002314                MOV     DESTRK,-(SP)    ;;PUSH DESTRK ON STACK
13 025322  012737  001167  002374        MOV     #631.,LOCYL     ;SET LO CYL BOUNDARY AND THE
14 025330  012737  177777  002372        MOV     #-1,HICYL       ;AND THE HI CYL BOUNDARY TO ILLEGAL LIMITS
15 025336  012704  036074                MOV     #DBUFF,R4       ;GET POINTER TO BEGINNING OF DATA BUFFER
16 025342  005003                        CLR     R3              ;R3 USED TO DETECT A TRACK MATCH
17 025344  012701  032464        1$:     MOV     #DEFBUF,R1      ;GET POINTER TO BEGINNING OF DEFECT BUFFER
18 025350  012105                2$:     MOV     (R1)+,R5        ;SAVE FIRST WORD (CYL ADRS)
19 025352  042705  170000                BIC     #170000,R5      ;STRIP THE STATUS BITS
20 025356  011146                        MOV     (R1),-(SP)      ;SAVE SECOND WORD (TRK/SEC ADRS)
21 025360  105016                        CLRB    (SP)            ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
22 025362  000316                        SWAB    (SP)            ;MAKE IT THE LOW BYTE OF DATA.
23 025364  022603                        CMP     (SP)+,R3        ;IS THERE A DEFECT ON THIS TRACK ?
24 025366  001415                        BEQ     5$              ;BR IF YES
25 025370  062701  000014        3$:     ADD     #<6*2>,R1       ;POINT TO NEXT DEFECT IN TABLE
26 025374  005711                        TST     (R1)            ;IS NEXT ENTRY VALID ?
27 025376  001403                        BEQ     4$              ;BR IF NO
28 025400  020127  036072                CMP     R1,#ENDDEF      ;END OF DEFECT BUFFER YET ?
29 025404  101761                        BLOS    2$              ;BR IF NO
30 025406  005203                4$:     INC     R3              ;GET NEXT TRACK
31 025410  020327  000037                CMP     R3,#31.         ;LAST TRACK YET ?
32 025414  101753                        BLOS    1$              ;BR IF NO
33 025416  000137  026126                JMP     21$             ;NO DEFECTS FOUND
34
35 025422  010137  002250        5$:     MOV     R1,TEMPA        ;SAVE THE CURRENT TRACK
36 025426  005205                        INC     R5              ;GET NEXT CYLINDER
37 025430  005761  000016        6$:     TST     16(R1)          ;IS IT VALID DATA ?
38 025434  001764                        BEQ     4$              ;BR IF NO
39 025436  016646  000016                MOV     16(SP),-(SP)    ;SAVE SECOND WORD (TRK/SEC ADRS)
40 025442  105016                        CLRB    (SP)            ;GET RID OF LOW BYTE (EXTRACT TRACK) AND
41 025444  000316                        SWAB    (SP)            ;MAKE IT THE LOW BYTE OF DATA.
42 025446  022603                        CMP     (SP)+,R3        ;IS THERE A DEFECT ON THIS TRACK ?
43 025450  001410                        BEQ     7$              ;BR IF YES
44 025452  062701  000016                ADD     #<7*2>,R1       ;GET THE NEXT ENTRY, PLEASE
45 025456  005711                        TST     (R1)            ;IS IT VALID DATA ?
46 025460  001752                        BEQ     4$              ;BR IF NO
47 025462  020127  036072                CMP     R1,#ENDDEF      ;END OF DEFECT BUFFER YET ?
48 025466  101760                        BLOS    6$              ;BR IF NO
49 025470  101346                        BHI     4$              ;BR IF YES
50 025472  016146  000014        7$:     MOV     14(R1),-(SP)    ;SAVE FIRST WORD (CYL ADRS)
51 025476  042716  170000                BIC     #170000,(SP)    ;STRIP THE STATUS BITS
52 025502  022605                        CMP     (SP)+,R5        ;IS THERE A DEFECT ON THIS CYLINDER ?
53 025504  001331                        BNE     3$              ;BR IF NO
54 025506  062701  000016                ADD     #<7*2>,R1       ;GET THE LAST LOOKED AT ENTRY
55 025512  005037  002366                CLR     TEMP1           ;THESE ARE USED AS DATA
```

```
                                                ;DEFECT ACCUMULATORS
                                                ;GET # OF DEFECT ENTRIES IN TRACK DESCRIPTOR
                                                ;GET THE ORIGINAL BUFFER ADDRESS OF THE SCRATCH
                                                ;MOVE UP TO THE FIRST DEFECT VALUE
                                                ;NEXT DEFECT, PLEASE
                                                ;ACCUMULATE THIS INFORATION
                                                ;FOR BOTH CYLINDERS WITH A COMMON TRACK
                                                ;IS THE DATA WITHIN TOLERANCE?
                                                ;IF SO, THESE WILL MATCH...
                                                ;TAKE BRANCH IF SO
                                                ;NEXT RECORD ENTRY, PLEASE
                                                ;TRY AGAIN, IF NOT 0
                                                ;RESTORE THE STACK
                                                ;NO MATCH, KEEP LOOKING

                                                ;BACK UP THE POINTERS TO THE LAST DEFECT
                                                ;WHICH DEFECT MATCHED?
                                                ;IF (SP) MATCHES 4, IT IS THE CURRENT ONE.
                                                ;AND TALLY THE EVENT
                                                ;KEEP GOING

                                                ;RESTORE THE STACK
                                                ;GET THE LO CYLINDER ADDRESS FOR THE SCRATCH
                                                ;STRIP THE MASK BITS
                                                ;IS THE CYLINDER ADDRESS > 5?
                                                ;IF SO, TAKE BRANCH
                                                ;SET UP FOR THE FIRST LEGAL CYLINDER
                                                ;AND KEEP GOING

                                                ;REDUCE THE LOWEST SCRATCH ADDRESS BY 5
                                                ;LEGAL ADDRESS?
                                                ;IF =>, YES
                                                ;CREATE THE LEGAL ADDRESS
                                                ;SET UP FOR THE DRIVER
                                                ;MOVE R1 TO CYL ADDRESS OF BEGINNING OF SCRATCH
                                                ;SET UP THE FOUND HIGHEST CYL ADDRESS OF THE SCRATCH
                                                ;DESTRK = 0.
                                                ;SET UP FOR THE DRIVER
                                                ;STRIP THE UNWANTED MASK
                                                ;GET THE HIGHEST FOUND VALUE
                                                ;LOOK FOR GAPS; IE HEAD BOUNCES
                                                ;GET THE LARGEST SCRATCH VALUE
                                                ;BACK UP THE POINTER TO BEGINNING OF THE DEFECT RECORD
                                                ;GET THE ORIGINAL SCRATCH DATA
                                                ;LOAD THE DATA
                                                ;GET THE STARTING CYLINDER ADDRESS

                                                ;;PUSH (R1) ON STACK
                                                ;STRIP THE UNWANTED MASK
                                                ;IN RANGE?
                                                ;TAKE BRANCH IF SO
                                                ;UPDATE THE SCRATCH INFORMATION
                                                ;FINISHED WITH THIS CYLINDER,
                                                ;GET THE NEXT CYLINDER IN THE SEQUENCE
                                                ;STILL IN RANGE?
                                                ;IF <=, YES
                                                ;GO BACK AND LOOK FOR ANOTHER SCRATCH
```

```
                112
026026  062701  000016          16$:    ADD     #<7*2>,R1       ;NEXT BUFFER ENTRY, PLEASE
026032  026637  177776  002372  17$:    CMP      2(SP),HICYL    ;STILL IN RANGE?
026040  101402                          BLOS    18$             ;IF SO, TAKE THE BRANCH
026042  000137  025406                  JMP     4$              ;GET THE NEXT TRACK AND LOOK FOR ANOTHER SCRATCH
                117
026046  126137  000003  002315  18$:    CMPB    3(R1),DESTRK+1  ;TRACK MATCH ?
026054  001402                          BEQ     19$             ;IF SO, LOG THE DATA
026056  103763                          BLO     16$             ;IF LOWER, KEEP MOVING THROUGH THE BUFFER
026060  101346                          BHI     14$             ;IF HIGHER, GET THE TD FROM THE DISK, AND UPDATE IT
026062  012705  000007          19$:    MOV     #7,R5           ;7 ITEMS / RECORD
026066  012124                  20$:    MOV     (R1)+,(R4)+     ;LOG THIS SCRATCH DATA
026070  005305                          DEC     R5              ;ONE LESS ITEM / THIS RECORD
026072  003375                          BGT     20$             ;KEEP GOING UNTIL R5 = 0
026074  162701  000016                  SUB     #<7*2>,R1       ;BACK R1 UP TO THE TOP OF THIS RECORD
026100  004737  025144                  JSR     PC,COMPRS       ;STRIP OUT THIS DEFECT, AND COMPRESS THE BUFFER
026104  005237  002312                  INC     DESCYL          ;NEXT DESIRED CYLINDER PLEASE
026110  005237  002374                  INC     LOCYL           ;ONE LESS CYLINDER TO-GO
026114  020127  036072                  CMP     R1,#ENDDEF      ;END OF DEFECT BUFFER YET ?
026120  101734                          BLOS    15$             ;BR IF NO
026122  000137  025350                  JMP     2$              ;GET THE NEXT CYLINDER ADDRESS
                133
026126  012701  032464          21$:    MOV     #DEFBUF,R1      ;GET POINTER TO BEGINNING OF DEFECT BUFFER
026132  012705  000007          22$:    MOV     #7,R5           ;SEVEN ENTRIES / RECORD
026136  012124                  23$:    MOV     (R1)+,(R4)+     ;LOG THE NON SCRATCH RELATED DATA
026140  005305                          DEC     R5              ;ONE LESS ENTRY THIS RECORD
026142  003375                          BGT     23$             ;DO UNTIL R5 = 0
026144  005711                          TST     (R1)            ;IS NEXT ENTRY VALID DATA ?
026146  001371                          BNE     22$             ;BR IF YES
026150  012637  002314                  MOV     (SP)+,DESTRK    ;;POP STACK INTO DESTRK
026154  012637  002312                  MOV     (SP)+,DESCYL    ;;POP STACK INTO DESCYL
026160  000207                          RTS     PC
```

```
 1
 2                                  ;;*****************************************************************************
 3                                  ;THIS MODULE IS USED BY THE SCANNER MODE OF OPERATION
 4                                  ;*
 5                                  ;INPUTS ARE:    FROM (R1)+,(R1)+,(R1)+,(R1)+,(R1)+,(R1)+,(R1)+
 6                                  ;
 7                                  ;OUTPUTS ARE:   DESCYL, DESTRK, TRACK DESCRIPTOR BUFFER CALLED 'TDWRD1',
 8                                  ;               UPDATE DEFECT BUFFER POINTER IN R1 AND TEST THE INTERNAL
 9                                  ;               DEFECT STATUS BEFORE RETURNING.
10                                  ;*
11                                  ;THIS MODULE IS CALLED BY: MAIN
12                                  ;;*****************************************************************************
13
14 026162                          LOADTD:
   026162  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
15 026164  012700  031414                   MOV     #TDWRD1,R0      ;GET POINTER TO BEGINNING OF TD BUFFER
16 026170  011137  002312                   MOV     (R1),DESCYL     ;LOAD THE DESIRED CYLINDER ADDRESS AND
17 026174  042737  170000  002312          BIC     #170000,DESCYL  ;STRIP OFF STATUS BITS.
18 026202  012120                          MOV     (R1)+,(R0)+     ;LOAD CYLINDER ADDRESS INTO TD WORD #1
19 026204  011137  002314                   MOV     (R1),DESTRK     ;LOAD THE TRACK ADDRESS AND
20 026210  105037  002314                   CLRB    DESTRK          ;THE SECTOR ADDRESS.
21 026214  012120                          MOV     (R1)+,(R0)+     ;LOAD TRACK/SECTOR ADRS INTO TD WORD #2
22 026216  012120                          MOV     (R1)+,(R0)+     ;LOAD TD WORD #3,
23 026220  012120                          MOV     (R1)+,(R0)+     ;TD WORD #4,
24 026222  012120                          MOV     (R1)+,(R0)+     ;TD WORD #5,
25 026224  012110                          MOV     (R1)+,(R0)      ;AND TD WORD #6.
26 026226  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
27 026230  032721  000400                   BIT     #BIT8,(R1)+     ;GET OVER INTERNAL STATUS WORD AND
28                                                                  ;TEST DEFECT STATUS BIT IN TD LIST.
29 026234  000207                          RTS     PC              ;TAKE THE RETURN NOW
30
31                                  ;;*****************************************************************************
32                                  ;THIS MODULE IS USED TO SAVE THE CONTENTS OF A BUFFER CALLED 'TDWRD1' IN THE
33                                  ;A BUFFER CALLED 'TDTMP1'.
34                                  ;*
35                                  ;INPUTS ARE:    TDWRD1 - TDWRD6
36                                  ;
37                                  ;OUTPUTS ARE:   TDTMP1 - TDTMP6
38                                  ;*
39                                  ;THIS MODULE IS CALLED BY: MAIN
40                                  ;;*****************************************************************************
41
42 026236                          SAVETD:
   026236  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   026240  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
43 026242  012700  031414                   MOV     #TDWRD1,R0      ;GET POINTER TO BEGINNING OF TD BUFFER
44 026246  012701  031444                   MOV     #TDTMP1,R1      ;GET POINTER TO BEGINNING OF TD TEMP. BUFFER
45 026252  012021                          MOV     (R0)+,(R1)+     ;SAVE TD WORD #1,
46 026254  012021                          MOV     (R0)+,(R1)+     ;TD WORD #2,
47 026256  012021                          MOV     (R0)+,(R1)+     ;TD WORD #3,
48 026260  012021                          MOV     (R0)+,(R1)+     ;TD WORD #4,
49 026262  012021                          MOV     (R0)+,(R1)+     ;TD WORD #5,
50 026264  011011                          MOV     (R0),(R1)       ;AND TD WORD #6.
51 026266  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
   026270  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
52 026272  000207                          RTS     PC
53
```

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

 026274                         RESTTD:
 026274  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 026276  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 026300  012700  031444                MOV     #TDTMP1,R0      ;GET POINTER TO BEGINNING OF TD TEMP. BUFFER
 026304  012701  031414                MOV     #TDWRD1,R1      ;GET POINTER TO BEGINNING OF TD BUFFER
 026310  012021                        MOV     (R0)+,(R1)+     ;SAVE TD WORD #1,
 026312  012021                        MOV     (R0)+,(R1)+     ;TD WORD #2,
 026314  012021                        MOV     (R0)+,(R1)+     ;TD WORD #3,
 026316  012021                        MOV     (R0)+,(R1)+     ;TD WORD #4,
 026320  012021                        MOV     (R0)+,(R1)+     ;TD WORD #5,
 026322  011011                        MOV     (R0),(R1)       ;AND TD WORD #6.
 026324  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 026326  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 026330  000207                        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
                                ;;*****************************************************************************

 026332                         DESTD:
 026332  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 026334  012700  031414                MOV     #TDWRD1,R0      ;POINT TO BEGINNING OF TD BUFFER
 026340  013710  002312                MOV     DESCYL,(R0)     ;LOAD THE CYLINDER ADDRESS AND
 026344  052720  030000                BIS     #BIT13!BIT12,(R0)+      ;MARK WORD AS TD & FMT16 IN TD WORD #1
 026350  013710  002314                MOV     DESTRK,(R0)     ;LOAD THE TRACK ADDRESS AND
 026354  052720  000377                BIS     #377,(R0)+      ;THE SECTOR ADDRESS IN TD WORD #2
 026360  012720  140000                MOV     #NULL,(R0)+     ;NULL TD WORD #3,
 026364  012720  140000                MOV     #NULL,(R0)+     ;TD WORD #4,
 026370  012720  140000                MOV     #NULL,(R0)+     ;TD WORD #5,
 026374  012710  140000                MOV     #NULL,(R0)      ;AND TD WORD #6.
 026400  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 026402  000207                        RTS     PC              ;TAKE THE RETURN NOW
```

```
 1
 2                                  ;;*****************************************************************************
 3                                  ;THIS MODULE IS USED TO RESTORE THE MASTER BUFFER 'DBUFF' BETWEEN ITERATIONS
 4                                  ;OF THE SCANNER.
 5                                  ;;*****************************************************************************
 6
 7 026404  012701  032464          RESTOR: MOV     #DEFBUF,R1      ;GET POINTER TO BEGINNING OF DEFECT BUFFER
 8 026410  012703  036074                  MOV     #DBUFF,R3       ;GET POINTER TO BEGINNING OF DATA BUFFER
 9 026414  005713                  1$:     TST     (R3)            ;IS THERE A DEFECT ENTRY HERE ?
10 026416  001430                          BEQ     5$              ;BR IF NO
11 026420  032763  001000  000014          BIT     #BIT9,14(R3)    ;WAS DEFECT AN INTERPOLATED SCRATCH ?
12 026426  001004                          BNE     2$              ;BR IF YES
13 026430  032763  000400  000014          BIT     #BIT8,14(R3)    ;WAS DEFECT BIT SET IN STATUS ?
14 026436  001006                          BNE     3$              ;BR IF YES
15 026440  062703  000016          2$:     ADD     #<7*2>,R3       ;DO NOT RESTORE ENTRY BACK TO DEFECT BUFFER
16 026444  020327  120072                  CMP     R3,#ENDBUF      ;DONE WITH BUFFER YET ?
17 026450  101761                          BLOS    1$              ;BR IF NO
18 026452  000412                          BR      5$
19
20 026454  012704  000007          3$:     MOV     #7,R4           ;SEVEN ENTRIES / RECORD
21 026460  012321                  4$:     MOV     (R3)+,(R1)+     ;RESTORE THE ORIGINAL BUFFER
22 026462  005304                          DEC     R4              ;ONE LESS ENTRY THIS RECORD
23 026464  003375                          BGT     4$              ;DO UNTIL R4 = 0
24 026466  010137  002240                  MOV     R1,LASLOC       ;SAVE THIS THE 1ST UNUSED LOCATION FOR NOW
25 026472  020127  036072                  CMP     R1,#ENDDEF      ;END OF DEFECT BUFFER YET ?
26 026476  101746                          BLOS    1$              ;BR IF NO
27 026500  000207                  5$:     RTS     PC
28
37
44
```

```
     12                                 .SBTTL  REPORT CODING SECTION
     40
     42                                 ;++
     43                                 ; THE REPORT CODING SECTION CONTAINS THE
     44                                 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
     45                                 ; -
     46
     47 026502                          L$RPT::
     48
     60
     61 026502  000167                          .WORD   J$JMP
        026504  000000                          .WORD   L10007 2-.
     62
     74                                         .EVEN
     75
     76 026506                          L10007:
        026506  104425                          TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

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

026510                          L$PROT::
026510  000000                          0               ;P TABLE OFFSET OF CSR
026512  177777                          -1              ;NOT A MASSBUS DEVICE
026514  000006                          6               ;P-TABLE OFFSET DRIVE #
```

```
                                .SBTTL  INITIALIZE SECTION

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

026516                          L$INIT::
                                                        ;RESET THE WORLD
026516  104433                          TRAP    C$RESET
                                                        ;POWER FAIL START ?
026520  012700  000034                  MOV     #EF.PWR,R0
026524  104447                          TRAP    C$REFG
                                                        ;GO TO 1$ IF NO
026526  103002                          BCC     1$
                                                        ;EXIT IF YES
026530  104432                          TRAP    C$EXIT
026532  001014                          .WORD   L10011-.
026534                          1$:                     ;CONTINUE COMMAND ?
026534  012700  000036                  MOV     #EF.CON,R0
026540  104447                          TRAP    C$REFG
                                                        ;GO TO 2$ IF NO
026542  103004                          BCC     2$
026544  105237  002247                  INCB    CONFLG  ;INDICATE CONTINUE COMMAND AND EXIT
026550  104432                          TRAP    C$EXIT
026552  000774                          .WORD   L10011-.
026554                          2$:                     ;RESTART COMMAND ?
026554  012700  000037                  MOV     #EF.RES,R0
026560  104447                          TRAP    C$REFG
                                                        ;GO TO 3$ IF YES
026562  103406                          BCS     3$
                                                        ;START COMMAND ?
026564  012700  000040                  MOV     #EF.STA,R0
026570  104447                          TRAP    C$REFG
                                                        ;GO TO 3$ IF YES
026572  103402                          BCS     3$
026574  000137  027260                  JMP     15$     ;GO TO 15$ IF NO

026600  005037  002114          3$:     CLR     L$TEST  ;CLEAR TEST NUMBER
026604  005227  177777          4$:     INC     #-1     ;FIRST TIME THRU HERE ?
026610  001020                          BNE     6$      ;BR IF NO
026612                          5$:                     ;PRINT THE HELP MESSAGE
026612  012746  004634                  MOV     #FRMT50,-(SP)
026616  012746  000001                  MOV     #1,-(SP)
026622  010600                          MOV     SP,R0
026624  104417                          TRAP    C$PNTF
026626  062706  000004                  ADD     #4,SP
                                                        ;PRINT THE HELP MESSAGE
026632  012746  004742                  MOV     #FRMT51,-(SP)
026636  012746  000001                  MOV     #1,-(SP)
026642  010600                          MOV     SP,R0
026644  104417                          TRAP    C$PNTF
026646  062706  000004                  ADD     #4,SP
026652                          6$:                     ;PRINT 'OPTIONS (6=HELP) (D)  0 ?'
026652  104443                          TRAP    C$GMAN
026654  000406                          BR      10000$
026656  002160                          .WORD   OPTION
```

```
        026660  000052                          .WORD   T$CODE
        026662  006254                          .WORD   MESG7
        026664  000007                          .WORD   7
        026666  000000                          .WORD   T$LOLIM
        026670  000007                          .WORD   T$HILIM
        026672                          10000$:
    43  026672  005037  002156                  CLR     ANYWHR          ;DEFAULT; ACCESS TO FE CYLINDER ONLY
    44  026676  013737  002214  002212          MOV     TMPCYL,MAXCYL   ;RESTORE MAXIMUM CYLINDER VALUE
    45  026704  005737  002160                  TST     OPTION          ;FORMAT OPTION ?
    46  026710  001413                          BEQ     7$              ;BR IF YES
    47  026712  023727  002160  000002          CMP     OPTION,#2       ;SCAN OPTION ?
    48  026720  001407                          BEQ     7$              ;BR IF YES
    49  026722  023727  002160  000004          CMP     OPTION,#4       ;MODIFY OPTION ?
    50  026730  001403                          BEQ     7$              ;BR IF YES
    51  026732  005237  002156                  INC     ANYWHR          ;ALLOW ACCESS TO ANYWHERE ON THE MEDIA
    52  026736  000417                          BR      8$
    53  026740                          7$:                             ;PRINT 'DO YOU WANT TO WRITE ANYWHERE ON MEDIA?'
    54  026740  104443                          TRAP    C$GMAN
        026742  000404                          BR      10001$
        026744  002156                          .WORD   ANYWHR
        026746  000130                          .WORD   T$CODE
        026750  006342                          .WORD   WRITMG
        026752  000001                          .WORD   1
        026754                          10001$:
    55  026754  005737  002156                  TST     ANYWHR          ;SEE IF ACCESS IS TO ANYWHERE ON THE MEDIA,
    56  026760  001406                          BEQ     8$              ;BRANCH IF NOT, ELSE
    57                                                                  ;TELL USER THAT DATA WILL BE DESTROYED.
    58                                                                  ;PRINT '! CUSTOMER DATA WILL BE OVERWRITTEN !
    59                                                                  ;       ------------------------------------ -
    60                                                                  ;       CONTINUE ?'
    61  026762  104443                          TRAP    C$GMAN
        026764  000404                          BR      10002$
        026766  002156                          .WORD   ANYWHR
        026770  000120                          .WORD   T$CODE
        026772  006411                          .WORD   WRSAFM
        026774  000001                          .WORD   1
        026776                          10002$:
    63  026776  023727  002160  000006  8$:     CMP     OPTION,#6       ;PRINT HELP MESSAGE ?
    64  027004  001702                          BEQ     5$              ;BR IF YES
    65  027006  023727  002160  000004          CMP     OPTION,#4       ;WHAT IS OPTION ?
    66  027014  002116                          BGE     14$             ;BRANCH IF MODIFY OR WRITE FE-2
    67                                                                  ;ELSE, OPTION IS FORMAT, VERIFY, SCAN OR LIST
    68  027016  005037  002206                  CLR     DRVPAR          ;DEFAULT TO 'N' FOR CHANGE PARAMETERS
    69                                                                  ;PRINT 'CHANGE DRIVE PARAMETERS (L) N ?'
    70  027022  104443                          TRAP    C$GMAN
        027024  000404                          BR      10003$
        027026  002206                          .WORD   DRVPAR
        027030  000130                          .WORD   T$CODE
        027032  006224                          .WORD   MESG18
        027034  000001                          .WORD   1
        027036                          10003$:
    71  027036  005737  002206                  TST     DRVPAR          ;CHANGE DRIVE PARAMETERS ?
    72  027042  001446                          BEQ     10$             ;BR IF NO
    74  027044  005737  002156                  TST     ANYWHR          ;DON'T ALLOW MIN/MAX CYLINDER TO BE CHANGED,
    75  027050  001423                          BEQ     9$              ;IF ACCESS IS TO FE CYLINDER ONLY, ELSE
    77                                                                  ;PRINT 'MIN CYL (D)  0 ?'
    78  027052  104443                          TRAP    C$GMAN
```

```
    027054  000406                          BR      10004$
    027056  002210                          .WORD   MINCYL
    027060  000052                          .WORD   T$CODE
    027062  006322                          .WORD   MESG15
    027064  001777                          .WORD   1777
    027066  000000                          .WORD   T$LOLIM
    027070  001166                          .WORD   T$HILIM
    027072                          10004$:
 79
 80 027072  104443                          TRAP    C$GMAN          ;PRINT 'MAX CYL (D)  630 ?'
    027074  000406                          BR      10005$
    027076  002212                          .WORD   MAXCYL
    027100  000052                          .WORD   T$CODE
    027102  006332                          .WORD   MESG16
    027104  001777                          .WORD   1777
    027106  000000                          .WORD   T$LOLIM
    027110  001166                          .WORD   T$HILIM
    027112                          10005$:
 82 027112  013737  002212  002214          MOV     MAXCYL,TMPCYL   ;SAVE MAXIMUM CYLINDER VALUE
 84 027120                          9$:                             ;PRINT 'MIN TRK (D)  0 ?'
 85 027120  104443                          TRAP    C$GMAN
    027122  000406                          BR      10006$
    027124  002216                          .WORD   MINTRK
    027126  000052                          .WORD   T$CODE
    027130  006302                          .WORD   MESG13
    027132  000037                          .WORD   37
    027134  000000                          .WORD   T$LOLIM
    027136  000037                          .WORD   T$HILIM
    027140                          10006$:
 86                                                                 ;PRINT 'MAX TRK (D)  31 ?'
 87 027140  104443                          TRAP    C$GMAN
    027142  000406                          BR      10007$
    027144  002220                          .WORD   MAXTRK
    027146  000052                          .WORD   T$CODE
    027150  006312                          .WORD   MESG14
    027152  000037                          .WORD   37
    027154  000000                          .WORD   T$LOLIM
    027156  000037                          .WORD   T$HILIM
    027160                          10007$:
 88
 89 027160  005737  002160          10$:    TST     OPTION          ;FORMAT OPTION ?
 90 027164  001006                          BNE     12$             ;BR IF NO
 91 027166                          11$:                            ;PRINT 'INHIBIT WRITE CHECK ?'
 92 027166  104443                          TRAP    C$GMAN
    027170  000404                          BR      10010$
    027172  002176                          .WORD   NOWRCK
    027174  000130                          .WORD   T$CODE
    027176  006544                          .WORD   MESG17
    027200  000001                          .WORD   1
    027202                          10010$:
 93
 94 027202  023727  002160  000002  12$:    CMP     OPTION,#2       ;SCAN OPTION ?
 95 027210  001006                          BNE     13$             ;BR IF NO
 96                                                                 ;PRINT 'DO YOU WANT TO RE-WRITE TD(S) WITH
 97                                                                 ;       NEW DEFECTS ?'
 98 027212  104443                          TRAP    C$GMAN
    027214  000404                          BR      10011$
```

```
        027216  002200                          .WORD   ENWTTD
        027220  000130                          .WORD   T$CODE
        027222  006642                          .WORD   MESG19
        027224  000001                          .WORD   1
        027226                          10011$:
     99
    100 027226  023727  002160  000003  13$:    CMP     OPTION,#3       ;LIST OPTION ?
    101 027234  001006                          BNE     14$             ;BR IF NO
    102                                                                 ;PRINT 'LIST HEADERS OF DEFECT SECTOR(S)?'
    103 027236  104443                          TRAP    C$GMAN
        027240  000404                          BR      10012$
        027242  002204                          .WORD   LISHDR
        027244  000130                          .WORD   T$CODE
        027246  006721                          .WORD   MESG23
        027250  000001                          .WORD   1
        027252                          10012$:
```

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

```
     56 027470  000411                          BR      27$
     57
     58                                         ;ENTER HERE TO VERIFY TD(S) & HEADER(S)
     59
     60 027472  012721  000075          25$:    MOV     #RDTD,(R1)+     ;LOAD THE READ TD OPERATION
     61 027476  012721  000073                  MOV     #RDHD,(R1)+     ;LOAD THE READ HEADER AND DATA FUNCTION
     62 027502  000411                          BR      29$
     63
     64                                         ;ENTER HERE TO MODIFY TD(S)
     65
     66 027504  005037  002416          26$:    CLR     MORETD          ;DEFAULT TO NO MORE TD'S TO MODIFY
     67                                                                 ;THE READ TD COMMAND IS IMBEDDED IN THE
     68                                                                 ;'INPUTD' SUBROUTINE
     69 027510  012721  000065                  MOV     #WRTTD,(R1)+    ;LOAD THE WRITE TD OPERATION
     70 027514  012721  000063          27$:    MOV     #FMTRK,(R1)+    ;LOAD THE FORMAT TRACK OPERATION
     71 027520  000402                          BR      29$
     72
     73                                         ;ENTER HERE TO LIST TD(S) WITH DEFECTS
     74
     75 027522  012721  000075          28$:    MOV     #RDTD,(R1)+     ;LOAD THE READ TD OPERATION
     76
     77                                         ;ENTER HERE TO SCAN FOR NEW DEFECTS
     78
     79 027526  022701  002612          29$:    CMP     #CMDQUE+10.,R1  ;QUEUE FULL?
     80 027532  103402                          BLO     30$             ;TAKE BRANCH IF SO
     81 027534  005021                          CLR     (R1)+           ;ZERO THE BALANCE OF THE QUEUE
     82 027536  000773                          BR      29$             ;AND DO UNTIL QUEUE IS DONE
     83 027540                          30$:
        027540  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
     84
    108
    109 027542  104432                          TRAP    C$EXIT
        027544  000002                          .WORD   L10011-.
    110
    122                                         .EVEN
    123
    124 027546                         L10011:
        027546  104411                          TRAP    C$INIT
```

```
                                .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.
                                ;--

027550                          L$AUTO::

027550                          L10012:
027550  104461                          TRAP    C$AUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

027552                          L$CLEAN::
                                                                ;SET PRIORITY TO 7
027552  012700  000340                  MOV     #PRI07,R0
027556  104441                          TRAP    C$SPRI
027560  012777  000040  152662          MOV     #CLR,@RPCS2     ;MASSBUS INIT TO CLEAR IMPENDING INTERRUPTS
027566  013777  002434  152654          MOV     DRVNO,@RPCS2    ;GET DRIVE NUMBER
                                                                ;CR-LF
027574  012746  002656                  MOV     #CRLF,-(SP)
027600  012746  000001                  MOV     #1,-(SP)
027604  010600                          MOV     SP,R0
027606  104417                          TRAP    C$PNTF
027610  062706  000004                  ADD     #4,SP
027614  005737  002114                  TST     L$TEST          ;DID PROGRAM ENTER TEST 1 ?
027620  001002                          BNE     1$              ;BR IF YES
027622  104432                          TRAP    C$EXIT
027624  000232                          .WORD   L10013 .

027626  012701  000001          1$:     MOV     #1,R1           ;R1 = 1
                                                                ;PRINT 'TRACKS WITH 1 DEFECTS= '
027632  013746  002302                  MOV     DEF1,-(SP)
027636  010146                          MOV     R1,-(SP)
027640  012746  003532                  MOV     #FRMT12,-(SP)
027644  012746  000003                  MOV     #3,-(SP)
027650  010600                          MOV     SP,R0
027652  104417                          TRAP    C$PNTF
027654  062706  000010                  ADD     #10,SP
027660  005201                          INC     R1              ;R1 = 2
                                                                ;PRINT 'TRACKS WITH 2 DEFECTS= '
027662  013746  002304                  MOV     DEF2,-(SP)
027666  010146                          MOV     R1,-(SP)
027670  012746  003532                  MOV     #FRMT12,-(SP)
027674  012746  000003                  MOV     #3,-(SP)
027700  010600                          MOV     SP,R0
027702  104417                          TRAP    C$PNTF
027704  062706  000010                  ADD     #10,SP
027710  005201                          INC     R1              ;R1 = 3
                                                                ;PRINT 'TRACKS WITH 3 DEFECTS= '
027712  013746  002306                  MOV     DEF3,-(SP)
027716  010146                          MOV     R1,-(SP)
027720  012746  003532                  MOV     #FRMT12,-(SP)
027724  012746  000003                  MOV     #3,-(SP)
027730  010600                          MOV     SP,R0
027732  104417                          TRAP    C$PNTF
027734  062706  000010                  ADD     #10,SP
027740  005737  002310                  TST     DEF4            ;WERE THERE ANY TRACKS WITH 4 DEFECTS ?
027744  001414                          BEQ     2$              ;BR IF NO
                                                                ;PRINT 'WARNING ** THERE ARE X. TRACK(S)
                                                                ;       WITH 4 DEFECTS'
027746  013746  002310                  MOV     DEF4,-(SP)
027752  012746  006206                  MOV     #WARN,-(SP)
```

```
CZRJKB0 RP07 FMTR/SCANNER MACRO V04.00 1 DEC 83 09:58:43 PAGE 74 1                                SEQ 0133
CLEANUP CODING SECTION

          027756  012746  005157                  MOV     #FRMT62, (SP)
          027762  012746  000003                  MOV     #3,-(SP)
          027766  010600                          MOV     SP,R0
          027770  104417                          TRAP    C$PNTF
          027772  062706  000010                  ADD     #10,SP
       33 027776                          2$:                                 ;PRINT 'TOTAL DEFECTS FOUND= X.'
       34 027776  013746  002254                  MOV     DEFCNT,-(SP)
          030002  012746  003600                  MOV     #FRMT13,-(SP)
          030006  012746  000002                  MOV     #2,-(SP)
          030012  010600                          MOV     SP,R0
          030014  104417                          TRAP    C$PNTF
          030016  062706  000006                  ADD     #6,SP
       35                                                                     ;PRINT 'DONE,  RETRIES MADE= X., ERRORS DETECTED= X'
       36 030022  013746  002266                  MOV     ERTTL,-(SP)
          030026  013746  002270                  MOV     ERRTTL,-(SP)
          030032  012746  003437                  MOV     #FRMT11,-(SP)
          030036  012746  000003                  MOV     #3,-(SP)
          030042  010600                          MOV     SP,R0
          030044  104417                          TRAP    C$PNTF
          030046  062706  000010                  ADD     #10,SP
       37
       46
       47 030052  104432                          TRAP    C$EXIT
          030054  000002                          .WORD   L10013-.
       48
       60                                         .EVEN
       61
       62 030056                          L10013:
          030056  104412                          TRAP    C$CLEAN
```

```
                                .SBTTL  DROP UNIT SECTION

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

030060                          L$DU::

                                        .WORD   J$JMP
030060  000167                  .WORD   L10014-2-.
030062  000000
                                        .EVEN

030064                          L10014:
030064  104453                         TRAP    C$DU
```

```
      1                                .SBTTL  ADD UNIT SECTION
      2
      3                                ;++
      4                                ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
      5                                ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
      6                                ;  TO THE TEST CYCLE.
      7                                ;-
      8
      9 030066                         L$AU::
     10
     19
     20 030066  000167                          .WORD   J$JMP
        030070  000000                          .WORD   L10015 2 .
     21
     33                                         .EVEN
     34
     35 030072                         L10015:
        030072  104452                          TRAP    C$AU
```

```
                                .SBTTL  HARDWARE TESTS
                                .SBTTL  TEST 1: FORMAT SECTION

                                ;++
                                ; TEST #1
                                ;--

030074                          T1::
030074  005037  002272                  CLR     ERRMSK          ;RESET NO ERROR STATUS,
030100  005037  002274                  CLR     RTYCNT          ;RESET RETRY COUNT TO 0 AND
030104  105037  002243                  CLRB    SCANIT          ;THE RESET SCAN ITERATION COUNT.
030110  012737  000005  002226          MOV     #5,ERRMAX       ;LOAD THE ERROR LIMIT
030116  004737  015630                  JSR     PC,DRVINI       ;INITIALIZE THE DEVICE
030122  105737  002247                  TSTB    CONFLG          ;DOING CONTINUE COMMAND ?
030126  001010                          BNE     1$              ;BR IF YES
030130  013737  002210  002312          MOV     MINCYL,DESCYL   ;LOAD THE MINIMUM DESIRED CYLINDER
030136  013737  002216  002314          MOV     MINTRK,DESTRK   ;LOAD THE MINIMUM DESIRED TRACK
030144  000337  002314                  SWAB    DESTRK          ;TRACK ADDRESS IS THE HIGH BYTE
030150  105037  002247          1$:     CLRB    CONFLG          ;CLEAR CONTINUE FLAG
030154  005737  002156                  TST     ANYWHR          ;SEE IF ACCESS IS TO ANYWHERE ON THE MEDIA,
030160  001007                          BNE     2$              ;BRANCH IF SO, ELSE
030162  013746  002222                  MOV     LSCYL,-(SP)     ;GET LAST USER CYLINDER ADDRESS AND
030166  005216                          INC     (SP)            ;MAKE THE FIRST FE CYLINDER.
030170  011637  002312                  MOV     (SP),DESCYL     ;SETUP THE DESIRED CYLINDER
030174  012637  002212                  MOV     (SP)+,MAXCYL    ;SETUP THE MAXIMUM CYLINDER
030200  005737  002262          2$:     TST     SOFSW           ;DID DRIVE INITIALIZE OK?
030204  001402                          BEQ     3$              ;BR IF YES
030206  000137  031126                  JMP     EXIT1           ;JUMP TO EXIT

030212  005737  002174          3$:     TST     WRTFE2          ;ARE WE WRITTING 2ND FE CYLINDER ?
030216  001404                          BEQ     4$              ;BR IF NO
030220  004737  021050                  JSR     PC,FETWO        ;GO FORMAT FE CYLINDER 2
030224  000137  031126                  JMP     EXIT1           ;JUMP TO EXIT

030230  004737  017714          4$:     JSR     PC,RDBSF        ;READ BAD SECTOR FILE (DEC 144 FILE)
030234  012702  031474                  MOV     #MFGBUF+8.,R2   ;GET POINTER TO BEGINNING OF MFG BSF BUFFER
                                                                ;TO GET OVER 2 S/N WRDS AND 2 ALL 0'S WRDS.
030240  010237  011626          5$:     MOV     R2,ENTBSF       ;UPDATE POINTER FOR NEXT ENTRY INTO BSF
030244  005712                          TST     (R2)            ;IS THERE ANY ROOM IN THE BSF ?
030246  100404                          BMI     6$              ;BR IF YES
030250  022222                          CMP     (R2)+,(R2)+     ;NEXT POINTER ENTRY, PLEASE
030252  020227  032462                  CMP     R2,#ENDBSF      ;END OF BUFFER YET ?
030256  103770                          BLO     5$              ;BR IF NO

030260                          6$:                             ;>>>>>>>>>>BREAK BACK TO MONITOR<<<<<<<<<<
030260  104422                          TRAP    C$BRK
030262  005737  002166                  TST     SCANR           ;SEE IF SCANNER MODE WAS ENABLED,
030266  001072                          BNE     SCAN            ;BRANCH IF YES, ELSE
                                                                ;MUST BE FORMAT, VERIFY OR MODIFY MODE.
030270  012705  002600                  MOV     #CMDQUE,R5      ;GET THE LIST OF COMMANDS
030274  022715  000075                  CMP     #RDTD,(R5)      ;IS THE COMMAND A READ TD ?
030300  001004                          BNE     7$              ;BR IF NO
030302  004737  021440                  JSR     PC,READTD       ;CALL THE READ TD SUBROUTINE
030306  000450                          BR      12$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
030310  000407                          BR      8$              ;ELSE RETURN HERE, NO ERROR

030312  021527  000065          7$:     CMP     (R5),#WRTTD     ;IS THE COMMAND A WRITE TD ?
```

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

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

```
                                        ;CORRESPONDING TRACK DESCRIPTOR(S) IF ENABLED.
                                        
030706  105737  002246          10$:    TSTB    NEWCNT          ;ANY NEW DEFECTS FOUND ?
030712  001475                          BEQ     19$             ;BR IF ZERO, NO
030714  012701  036074          11$:    MOV     #DBUFF,R1       ;GET POINTER TO BEGINNING OF DATA BUFFER AND
030720  005021                  12$:    CLR     (R1)+           ;INITIALIZE IT.
030722  020127  120072                  CMP     R1,#ENDBUF      ;DONE BUFFER YET ?
030726  101774                          BLOS    12$             ;BR IF NO
030730  004737  025312                  JSR     PC,FINSCR       ;LOOK FOR SCRATCHES IN THIS AREA;
                                                                ;PUT THE CONTENTS OF 'DEFBUF' (INCLUDING ANY
                                                                ;INTERPOLATED SCRATCHES) INTO 'DBUFF'.
030734  012701  036074                  MOV     #DBUFF,R1       ;GET POINTER TO BEGINNING OF DATA BUFFER
030740  004737  022224                  JSR     PC,NEWLST       ;PRINT NEW DEFECTS, IF ANY
030744  005737  002200                  TST     ENWTTD          ;IS WRITE TD(S) ENABLED ?
030750  001456                          BEQ     19$             ;BR IF NO
030752  013700  002226          13$:    MOV     ERRMAX,R0       ;SETUP RETRY COUNT
030756  004737  026162                  JSR     PC,LOADTD       ;LOAD THE TD FOR THE CORRECT TRACK AND
030762  001426                          BEQ     17$             ;BRANCH IF NO DEFECT STATUS
030764  004737  026236                  JSR     PC,SAVETD       ;SAVE TD WRITE BUFFER JUST IN CASE THERE
                                                                ;IS RETRY ATTEMPT.
030770  004737  026274          14$:    JSR     PC,RESTTD       ;NOW RESTORE THE TD BUFFER
030774  004737  021152                  JSR     PC,WRITTD       ;CALL THE WRITE TD SUBROUTINE
031000  000417                          BR      17$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE, NO ERROR
031002  004737  021440                  JSR     PC,READTD       ;CALL THE READ TD SUBROUTINE
031006  000401                          BR      15$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
031010  000403                          BR      16$             ;ELSE RETURN HERE, NO ERROR
031012  005300                  15$:    DEC     R0              ;DID WE EXCEED RETRY ?
031014  001411                          BEQ     17$             ;BR IF YES
031016  000764                          BR      14$             ;TRY WRITTING TD AGAIN

031020  012737  000063  002404  16$:    MOV     #FMTRK,FUNCTN   ;LOAD THE FORMAT COMMAND FOR THE DRIVER
031026  004737  015452                  JSR     PC,MAKTRK       ;NOW EXECUTE THE COMMAND
031032  000402                          BR      17$             ;RETURN HERE IF EXCEEDED RETRY LIMIT
                                                                ;ELSE RETURN HERE, NO ERROR
031034  004737  020676                  JSR     PC,CNTDEF       ;COUNT THESE NEW DEFECTS IN THE OVERALL COUNTER
031040  005711                  17$:    TST     (R1)            ;IS THIS END OF DEFECTS IN BUFFER ?
031042  001403                          BEQ     18$             ;BR IF YES
031044  020127  120072                  CMP     R1,#ENDBUF      ;AT END OF THE SCRATCH BUFFER ?
031050  101740                          BLOS    13$             ;BR IF NO

                                        ;NOW GET THE TRACK DESCRPTORS JUST WRITTEN AND GO SCAN THEM FOR
                                        ;ANYMORE POSSIBLE DEFECTS AFTER THE CORRECTION.

031052  105037  002246          18$:    CLRB    NEWCNT          ;RESET THE LOCAL DEFECT COUNTER
031056  004737  026404                  JSR     PC,RESTOR       ;NOW RESTORE THE ORIGINAL DEFECT BUFFER;
                                                                ;PUT THE CONTENTS OF 'DBUFF' (EXCLUDING THE
                                                                ;INTERPOLATED SCRATCHES) INTO 'DEFBUF'.
031062  012737  032464  002256          MOV     #DEFBUF,TBLPTR  ;RESTORE POINTER TO BEGINNING OF DEFECT BUFFER
031070  004737  020604                  JSR     PC,GETNEX       ;GET NEXT DEFECT TO CHECK
031074  112737  177777  002242          MOVB    #-1,FIRPAS      ;INDICATE 2ND PASS OF SCANNER
031102  000137  030536                  JMP     3$              ;DO SOME MORE SCANNING

031106  023737  002312  002212  19$:    CMP     DESCYL,MAXCYL   ;DONE ALL CYLINDERS YET ?
031114  101002                          BHI     20$             ;BR IF YES
031116  000137  030454                  JMP     SCAN            ;DO SOME MORE SCANNING
```

```
    115 031122  004737  020160          20$:    JSR     PC,WRTBSF       ;WRITE THE BAD SECTOR FILE
    116
    117 031126                          EXIT1:
    123
    135                                         .EVEN
    136
    137 031126                          L10016:
        051126  104401                          TRAP    C$ETST
    138
```

```
 2                                      .TITLE PARAMETER CODING
13
14                                      .SBTTL  HARDWARE PARAMETER CODING SECTION
42
44                                      ;++
45                                      ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
46                                      ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
47                                      ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
48                                      ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
49                                      ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
50                                      ; WITH THE OPERATOR.
51                                      ;--
52
53 031130  000022                               .WORD L10017-L$HARD/2
   031132                               L$HARD::
54                                                                      ;PRINT 'RPCS1 ADRS?'
55 031132  000031                               .WORD   T$CODE
   031134  031176                               .WORD   MESG1
   031136  160000                               .WORD   T$LOLIM
   031140  177777                               .WORD   T$HILIM
56                                                                      ;PRINT 'VECTOR ADRS?'
57 031142  001031                               .WORD   T$CODE
   031144  031211                               .WORD   MESG4
   031146  000000                               .WORD   T$LOLIM
   031150  000377                               .WORD   T$HILIM
58                                                                      ;PRINT 'BR LEVEL?'
59 031152  002032                               .WORD   T$CODE
   031154  031225                               .WORD   MESG5
   031156  000340                               .WORD   340
   031160  000000                               .WORD   T$LOLIM
   031162  000007                               .WORD   T$HILIM
60                                                                      ;PRINT 'DRIVE #?'
61 031164  003032                               .WORD   T$CODE
   031166  031236                               .WORD   MESG6
   031170  000007                               .WORD   7
   031172  000000                               .WORD   T$LOLIM
   031174  000007                               .WORD   T$HILIM
62                                              .EVEN
   031176                               L10017:
63
67 031176      122     120     103  MESG1:  .ASCIZ  /RPCS1 ADRS/
68 031211      126     105     103  MESG4:  .ASCIZ  /VECTOR ADRS/
69 031225      102     122     040  MESG5:  .ASCIZ  /BR LEVEL/
70 031236      104     122     111  MESG6:  .ASCIZ  /DRIVE #/
71
72                                      .EVEN
76
86
```

```
      1                                 .SBTTL  SOFTWARE PARAMETER CODING SECTION
      2
      3                                 ;++
      4                                 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
      5                                 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
      6                                 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
      7                                 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
      8                                 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
      9                                 ; WITH THE OPERATOR.
     10                                 ;--
     11
     12 031246  000000                          .WORD L10020-L$SOFT/2
        031250                          L$SOFT::
     13
     22                                         .EVEN
        031250                          L10020:
     23
     30
     31 031250                          $PATCH::.BLKW   50.                 ;PROGRAM PATCH AREA (50. WORDS)
     32
     38 031414  000000                  TDWRD1::.WORD   0                   ;TD WORD #1
        031416  000000                  TDWRD2::.WORD   0                   ;TD WORD #2
        031420  000000                  TDWRD3::.WORD   0                   ;TD WORD #3
        031422  000000                  TDWRD4::.WORD   0                   ;TD WORD #4
        031424  000000                  TDWRD5::.WORD   0                   ;TD WORD #5
        031426  000000                  TDWRD6::.WORD   0                   ;TD WORD #6
     42
     48 031430  000000                  TDCPY1::.WORD   0                   ;COPY OF TD WORD #1
        031432  000000                  TDCPY2::.WORD   0                   ;COPY OF TD WORD #2
        031434  000000                  TDCPY3::.WORD   0                   ;COPY OF TD WORD #3
        031436  000000                  TDCPY4::.WORD   0                   ;COPY OF TD WORD #4
        031440  000000                  TDCPY5::.WORD   0                   ;COPY OF TD WORD #5
        031442  000000                  TDCPY6::.WORD   0                   ;COPY OF TD WORD #6
     52
     58 031444  000000                  TDTMP1::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #1
        031446  000000                  TDTMP2::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #2
        031450  000000                  TDTMP3::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #3
        031452  000000                  TDTMP4::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #4
        031454  000000                  TDTMP5::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #5
        031456  000000                  TDTMP6::.WORD   0                   ;TEMPORARY STORAGE FOR TD WORD #6
     62
     63              031460             BEGBUF  = .                         ;BEGINNING OF BUFFER SECTION
     64 031460                          TMPBSF::.BLKW   2                   ;GENERAL BSF USE
     65 031464                          MFGBUF::.BLKW   256.                ;MFG BUFFER FOR DEC 144 FILE
     66              032462             ENDBSF  = .-2                       ;END OF BSF BUFFER
     67
     68              032464             DEFBUF  = .                         ;BEGINNING OF DEFECT BUFFER (SCAN MODE)
     69 032464                          HDRBLK::.BLKW   6*50.               ;HEADER BLOCK (6 WRDS X 50. SECTORS)
     70 033614                          TDSBLK::.BLKW   6*50.               ;TRACK DESCRIPTOR BLOCK        "
     71 034744                          LINBUF::.BLKW   6*50.               ;BEGINNING OF LINEAR BUFFER    "
     72              036072             ENDLIN  = .-2                       ;END OF LINEAR BUFFER
     73              036072             ENDDEF  = ENDLIN                    ;END OF DEFECT BUFFER
     74
     75 036074                          DBUFF:: .BLKW   256.*50.            ;DATA BUFFER (256. WRDS X 50. SECTORS)
     76              120072             ENDBUF  = .-2                       ;END OF BUFFER SECTION
     77
     84                                         .EVEN
```

```
        120074  120114                          .WORD T$FREE
        120076  000006                          .WORD T$SIZE
        1201C0                          L$LAST::
```

```
      1
     14
     16 120100  000000                          .WORD   0
        120102  000004                          .WORD   L10023 ./2 1
        120104                          L10021:
     17 120104  176700                          .WORD   176700
     18 120106  000254                          .WORD   254
     19 120110  000240                          .WORD   240
     20 120112  000000                          .WORD   0
     21 120114                          L10023:
     23         000001                          .END
```

SYMBOL TABLE

```
ADR    = 000020 G      CMDQUE  002600 G      C$RESE= 000033          ECI    = 004000 G      FINDWR  024040
ANDMDR   006103 G      CMOD   = 100000 G      C$REVI= 000003          EF.CON= 000036 G      FINSCR  025312
ANYWHR   002156 G      CNTDEF  020676         C$RFLA= 000021          EF.NEW= 000035 G      FIRPAS  002242 G
AOE    = 001000 G      COMPRS  025144         C$RPT = 000025          EF.PWR= 000034 G      FMTRK  = 000063 G
ASSEMB= 000010         CONFLG  002247 G       C$SEFG= 000046          EF.RES= 000037 G      FMT16  = 010000 G
ATA    = 100000 G      CONTLT  002362 G       C$SPRI= 000041          EF.STA= 000040 G      FORMAT  005700 G
ATO    = 000001 G      CPE    = 040000 G      C$SVEC= 000037          EM1      006766 G      FORMT   002162 G
AT1    = 000002 G      CPU    = 020000 G      C$TPRI= 000013          EM10     007215 G      FRMTT   002661 G
AT2    = 000004 G      CR     = 000015 G      DASCAN  023406          EM11     007243 G      FRMT00  002664 G
AT3    = 000010 G      CRE144  020550         DATA   = 000402 G      EM12     007267 G      FRMT01  002755 G
AT4    = 000020 G      CRLF    002656 G       DBUFF    036074 G      EM13     007326 G      FRMT02  003035 G
AT5    = 000040 G      CURENT  006030 G       DCK    = 100000 G      EM14     007351 G      FRMT03  003126 G
AT6    = 000100 G      C$AU   = 000052        DCU    = 000040 G      EM15     007377 G      FRMT04  003206 G
AT7    = 000200 G      C$AUTO= 000061         DEFBUF= 032464          EM16     007454 G      FRMT05  003300 G
A16    = 000400 G      C$BRK  = 000022        DEFCNT  002254 G       EM17     007531 G      FRMT06  003362 G
A17    = 001000 G      C$BSEG= 000004         DEFSEC  002406 G       EM2      007013 G      FRMT07  003402 G
BAI    = 000010 G      C$BSUB= 000002         DEFTRK  002410 G       EM20     007576 G      FRMT10  003421 G
BEGBUF= 031460         C$CEFG= 000045         DEF1    002302 G       EM21     007654 G      FRMT11  003437 G
BELL   = 000007 G      C$CLCK= 000062         DEF2    002304 G       EM22     007677 G      FRMT12  003532 G
BGNPAT= 002142 G       C$CLEA= 000012         DEF3    002306 G       EM3      007037 G      FRMT13  003600 G
BIT0   = 000001 G      C$CLOS= 000035         DEF4    002310 G       EM4      007063 G      FRMT14  003640 G
BIT00  = 000001 G      C$CLP1= 000006         DELTA   002570 G       EM5      007123 G      FRMT15  003651 G
BIT01  = 000002 G      C$CVEC= 000036         DESCYL  002312 G       EM6      007146 G      FRMT16  003661 G
BIT02  = 000004 G      C$DCLN= 000044         DESTD   026332         EM7      007164 G      FRMT17  003754 G
BIT03  = 000010 G      C$DODU= 000051         DESTRK  002314 G       ENDBSF= 032462         FRMT20  004034 G
BIT04  = 000020 G      C$DRPT= 000024         DFPTBL  002130 G       ENDBUF= 120072         FRMT22  004070 G
BIT05  = 000040 G      C$DU   = 000053        DIAG   = 000035 G      ENDDEF= 036072         FRMT23  004112 G
BIT06  = 000100 G      C$EDIT= 000003         DIAGMC= 000000         ENDLIN= 036072         FRMT24  004133 G
BIT07  = 000200 G      C$ERDF= 000055         DLT    = 100000 G      ENDPAT= 002156 G       FRMT30  004203 G
BIT08  = 000400 G      C$ERHR= 000056         DMD    = 100000 G      ENDPTR  002230 G       FRMT31  004231 G
BIT09  = 001000 G      C$ERRO= 000060         DMPREG  011132         ENDTAB  002232 G       FRMT32  004253 G
BIT1   = 000002 G      C$ERSF= 000054         DPE    = 000010 G      ENTBSF  011626         FRMT33  004351 G
BIT10  = 002000 G      C$ERSO= 000057         DPR    = 000400 G      ENWTTD  002200 G       FRMT35  004406 G
BIT11  = 004000 G      C$ESCA= 000010         DRIVER  010502         ERR    = 040000 G      FRMT36  004515 G
BIT12  = 010000 G      C$ESEG= 000005         DRQ    = 004000 G      ERRFNC  002260 G       FRMT37  004565 G
BIT13  = 020000 G      C$ESUB= 000003         DRT0   = 000001 G      ERRMAX  002226 G       FRMT50  004634 G
BIT14  = 040000 G      C$ETST= 000001         DRT1   = 000002 G      ERRMSK  002272 G       FRMT51  004742 G
BIT15  = 100000 G      C$EXIT= 000032         DRT2   = 000004 G      ERRORS  013402         FRMT60  005021 G
BIT2   = 000004 G      C$GETB= 000026         DRT3   = 000010 G      ERRTTL  002270 G       FRMT61  005055 G
BIT3   = 000010 G      C$GETW= 000027         DRT4   = 000020 G      ERRVEC= 000004         FRMT62  005157 G
BIT4   = 000020 G      C$GMAN= 000043         DRT5   = 000040 G      ERR0     007722 G      FRMT63  005241 G
BIT5   = 000040 G      C$GPHR= 000042         DRT6   = 000100 G      ERR1     007754 G      FRMT64  005342 G
BIT6   = 000100 G      C$GPLO= 000030         DRT7   = 000200 G      ERR2     010076 G      FRMT65  005443 G
BIT7   = 000200 G      C$GPRI= 000040         DRT8   = 000400 G      ERR3     010220 G      FRMT66  005534 G
BIT8   = 000400 G      C$INIT= 000011         DRVCLR= 000011 G       ERR5     010246 G      FUNCTN  002404 G
BIT9   = 001000 G      C$INLP= 000020         DRVINI  015630         ERTTL    002266 G      F$AU   = 000015
BOE    = 000400 G      C$MANI= 000050         DRVNO   002434 G       EVL    = 000004 G      F$AUTO= 000020
BSE    = 100000 G      C$MEM  = 000031        DRVPAR  002206 G       EWN    = 000002 G      F$BGN  = 000040
BUFINI  016304         C$MSG  = 000023        DRVRDY  005757 G       EXIT1    031126         F$CLEA= 000007
BUSADR  002234 G       C$OPEN= 000034         DRVSN   002436 G       EXPCTD   002276 G      F$DU   = 000016
CALCTD  014326         C$PNTB= 000014         DRY    = 000200 G      EXPTD    006002 G      F$END  = 000041
CHABUF  011630         C$PNTF= 000017         DS     = 000154 G      E$END  = 002100         F$HARD= 000004
CHANGE  006170 G       C$PNTS= 000016         DSE    = 020000 G      E$LOAD= 000035         F$HW   = 000013
CHEKTD  024424         C$PNTX= 000015         DTE    = 010000 G      FER    = 000020 G      F$INIT= 000006
CHKBSF  015172         C$QIO  = 000377        DVA    = 004000 G      FETWO    021050         F$JMP  = 000050
CHKSEG  014116         C$RDBU= 000007         DVC    = 000200 G      FINDBA   024474         F$MOD  = 000000
CLR    = 000040 G      C$REFG= 000047         ECH    = 000100 G      FINDSE   024116         F$MSG  = 000011
```

```
F$PROT= 000021          ILF    = 000001 G      L$DEVP  002060 G      L10020  031250          NOGOOD  016276
F$PWR  = 000017          ILR    = 000002 G      L$DISP  002124 G      L10021  120104          NOOP   = 000001 G
F$RPT  = 000012          INPUTD  012006         L$DLY   002116 G      L10023  120114          NORUN   016272
F$SEG  = 000003          INSERT  016616         L$DTP   002040 G      MAKTRK  015452          NOWRCK  002176 G
F$SOFT= 000005          INTEN  = 000100 G       L$DTYP  002034 G      MAPSEC  016466          NULL   = 140000 G
F$SRV  = 000010          INTLEV  002244 G       L$DU    030060 G      MAXCYL  002212 G        OFFDIR= 000200 G
F$SUB  = 000002          IR     = 000100 G      L$DUT   002072 G      MAXSEG  002414 G        OFFSET  002236 G
F$SW   = 000014          ISR    = 000100 G      L$DVTY  002614 G      MAXTRK  002220 G        OM     = 000001 G
F$TEST= 000001          ITCNT   002264 G       L$EF    002052 G      MAXVAL  024564          ONEFIL= 000001
F1     = 000002 G        IXE    = 004000 G      L$ENVI  002044 G      MCPE   = 020000 G       OPI    = 020000 G
F2     = 000004 G        IXU    = 000100 G      L$ETP   002102 G      MERGE   015242 G        OPTION  002160 G
F3     = 000010 G        I$AU   = 000041        L$EXP1  002046 G      MESG1   031176          OR     = 000200 G
F4     = 000020 G        I$AUTO= 000041         L$EXP4  002064 G      MESG13  006302 G        O$APTS= 000000
F5     = 000040 G        I$CLN  = 000041        L$EXP5  002066 G      MESG14  006312 G        O$AU   = 000000
GAPVAL  024702           I$DU   = 000041        L$HARD  031132 G      MESG15  006322 G        O$BGNR= 000000
GETNEX  020604           I$HRD  = 000041        L$HIME  002120 G      MESG16  006332 G        O$BGNS= 000000
GETPAT  024262           I$INIT= 000041         L$HPCP  002016 G      MESG17  006544 G        O$DU   = 000000
GO     = 000001 G        I$MOD  = 000041        L$HPTP  002022 G      MESG18  006224 G        O$ERRT= 000000
G$CNTO= 000200           I$MSG  = 000041        L$HW    002130 G      MESG19  006642 G        O$GNSW= 000000
G$DELM= 000372           I$PROT= 000040         L$ICP   002104 G      MESG22  006570 G        O$POIN= 000001
G$DISP= 000003           I$PTAB= 000041         L$INIT  026516 G      MESG23  006721 G        O$SETU= 000001
G$EXCP= 000400           I$PWR  = 000041        L$LADP  002026 G      MESG38  012654          PAR    = 000010 G
G$HILI= 000002           I$RPT  = 000041        L$LAST  120100 G      MESG39  012675          PARTBL  002336 G
G$LOLI= 000001           I$SEG  = 000041        L$LOAD  002100 G      MESG4   031211          PAT    = 000020 G
G$NO   = 000000          I$SETU= 000041         L$LUN   002074 G      MESG40  012713          PATTRN  002326 G
G$OFFS= 000400           I$SFT  = 000041        L$MREV  002050 G      MESG41  012732          PGE    = 100000 G
G$OFSI= 000376           I$SRV  = 000041        L$NAME  002000 G      MESG42  012753          PGM    = 001000 G
G$PRMA= 000001           I$SUB  = 000041        L$PRIO  002042 G      MESG43  012772          PHF    = 000400 G
G$PRMD= 000002           I$TST  = 000041        L$PROT  026510 G      MESG44  013004          PIP    = 020000 G
G$PRML= 000000           J$JMP  = 000167        L$PRT   002112 G      MESG45  013016          PLTRK   002323 G
G$RADA= 000140           LASLOC  002240 G       L$REPP  002062 G      MESG46  013030          PNT    = 001000 G
G$RADB= 000000           LBC    = 002000 G      L$REV   002010 G      MESG47  013042          PRI    = 002000 G
G$RADD= 000040           LBT    = 002000 G      L$RPT   026502 G      MESG5   031225          PRI00  = 000000 G
G$RADL= 000120           LCE    = 001000 G      L$SOFT  031250 G      MESG6   031236          PRI01  = 000040 G
G$RADO= 000020           LF     = 000012 G      L$SPC   002056 G      MESG7   006254 G        PRI02  = 000100 G
G$XFER= 000004           LINBUF  034744 G       L$SPCP  002020 G      MFGBUF  031464 G        PRI03  = 000140 G
G$YES  = 000010          LISHDR  002204 G       L$SPTP  002024 G      MINCYL  002210 G        PRI04  = 000200 G
G1     = 000100 G        LIST    002172 G       L$STA   002030 G      MINSEG  002412 G        PRI05  = 000240 G
G2     = 000045 G        LISTDS  021714         L$SW    002142 G      MINTRK  002216 G        PRI06  = 000300 G
G3     = 000040 G        LOADTD  026162         L$TEST  002114 G      MODBY   002202 G        PRI07  = 000340 G
G4     = 000045 G        LOCYL   002374 G       L$TIML  002014 G      MODTD   002170 G        PSEL   = 002000 G
HCE    = 000200 G        LODBSF  011476         L$UNIT  002012 G      MOH    = 020000 G       RCVED   006015 G
HCI    = 002000 G        LOE    = 040000 G      L10000  002140        MOL    = 010000 G       RDBSF   017714
HCRC   = 000400 G        LOT    = 000010 G      L10001  002142        MOLINE  005727 G        RDDAT  = 000071 G
HDR    = 000010 G        LSCYL   002222 G       L10002  007752        MORETD  002416 G        RDHD   = 000073 G
HDRBLK  032464 G         LSTRK   002224 G       L10003  010074        MPE    = 000400 G       RDTD   = 000075 G
HDRO   = 000010 G        L$ACP   002110 G       L10004  010216        MSEG   = 000022 G       RDY    = 000200 G
HDSCAN  022732           L$APT   002036 G       L10005  010244        MSRDTD  005615 G        READIN= 000021 G
HELP   = 000000          L$AU    030066 G       L10006  010326        MSWRTD  005604 G        READTD  021440
HICYL   002372 G         L$AUT   002070 G       L10007  026506        MTD    = 040000 G       READY   010472
HIVALU  025072           L$AUTO  027550 G       L10011  027546        MXF    = 001000 G       RECAL  = 000007 G
HOE    = 100000 G        L$CCP   002106 G       L10012  027550        NBA    = 100000 G       RECALI  005672 G
IAE    = 002000 G        L$CLEA  027552 G       L10013  030056        NED    = 010000 G       RECVED  002300 G
IBE    = 010000 G        L$CO    002032 G       L10014  030064        NEGWRD  002376 G        REG     002514 G
IDU    = 000040 G        L$DEPO  002011 G       L10015  030072        NEM    = 004000 G       RELSE  = 000013 G
IER    = 020000 G        L$DESC  002622 G       L10016  031126        NEWCNT  002246 G        RELTIV  016536
ILEV   = 000004 G        L$DESP  002076 G       L10017  031176        NEWLST  022224          RELWRD  002400 G
```

```
RESET   013256          SAVTRK  002320 G        TDCPY2  031432 G        T$FREE= 120114          T1      030074 G
RESTOR  026404          SBE     000004 G        TDCPY3  031434 G        T$GMAN= 000000          UAM     000200 G
RESTTD  026274          SC      100000 G        TDCPY4  031436 G        T$HILI= 000007          UNIT    002420 G
RETRY   013304          SCAN    030454          TDCPY5  031440 G        T$LAST= 000001          UNS     040000 G
RHDATA  005713 G        SCANIT  002243 G        TDCPY6  031442 G        T$LOLI= 000000          UPDSCR  025174
RHEXT   002430 G        SCANR   002166 G        TDDEF   006046 G        T$LSYM= 010000          UPE     020000 G
RHTYPE  002432 G        SCF     000002 G        TDDUMP  017240          T$LTNO= 000001          US1     000001 G
RMR     000004 G        SC1     000100 G        TDLMTS  002352 G        T$NEST= 177777          US2     000002 G
RPADR   002422 G        SC16    002000 G        TDSBLK  033614 G        T$NSO = 000000          US4     000004 G
RPAS    002456 G        SC2     000200 G        TDSCAN  022546          T$NS1 = 000005          VERIFY  017614
RPBA    002444 G        SC32    004000 G        TDTMP1  031444 G        T$PCNT= 000000          VRIFY   002164 G
RPBAE   002510 G        SC4     000400 G        TDTMP2  031446 G        T$PTAB= 010022          VV      000100 G
RPCC    002476 G        SC64    010000 G        TDTMP3  031450 G        T$PTHV= 000001          WAIT    013106
RPCS1   002440 G        SC8     001000 G        TDTMP4  031452 G        T$PTNU= 000001          WARN    006206 G
RPCS2   002450 G        SDF     000020 G        TDTMP5  031454 G        T$SAVL= 177777          WCE     040000 G
RPCS3   002512 G        SEARCH= 000031 G        TDTMP6  031456 G        T$SEGL= 177777          WCF     000040 G
RPDA    002446 G        SECADD  002402 G        TDVALU  002332 G        T$SIZE= 000006          WCKD    000051 G
RPDB    002462 G        SEC50   002322 G        TDWRD1  031414 G        T$SUBN= 000000          WCKDAT  005636 G
RPDC    002474 G        SEEK    000005 G        TDWRD2  031416 G        T$TAGL= 177777          WCKHD   000053 G
RPDS    002452 G        SEIZE   013140          TDWRD3  031420 G        T$TAGN= 010024          WCKHDR  005653 G
RPDT    002466 G        SFPTBL  002142 G        TDWRD4  031422 G        T$TEMP= 000000          WLE     004000 G
RPEC1   002504 G        SIZE70  010330          TDWRD5  031424 G        T$TEST= 000001          WLOCK   005743 G
RPEC2   002506 G        SKI     040000 G        TDWRD6  031426 G        T$TSTM= 177777          WOR     001000 G
RPER1   002454 G        SOFSW   002262 G        TEMPA   002250 G        T$TSTS= 000001          WRDAT   005625 G
RPER2   002500 G        SORT    020414          TEMP1   002366 G        T$$AU = 010015          WRITCK  011734
RPER3   002502 G        SPIRAL  020646          TEMP2   002370 G        T$$AUT= 010012          WRITMG  006342
RPLA    002460 G        SUPRSS  002252 G        TEXT    002360 G        T$$CLE= 010013          WRITTD  021152
RPMR1   002464 G        SVCGBL= 000000          TKWCNT  002364 G        T$$DAT= 010023          WRL     004000 G
RPOF    002472 G        SVCINS= 000000          TMDLTA  015170          T$$DU = 010014          WRSAFM  006411
RPSN    002470 G        SVCSUB= 000000          TMPBSF  031460 G        T$$HAR= 010017          WRTBSF  020160
RPVEC   002424 G        SVCTAG= 000000          TMPCYL  002214 G        T$$HW = 010000          WRTDAT= 000061 G
RPWC    002442 G        SVCTST= 000000          TOLER   024150          T$$INI= 010011          WRTFE2  002174 G
RTYCNT  002274 G        S$LSYM= 010000          TOLRAN  024336          T$$MSG= 010006          WRTTD   000065 G
RWU1    002000 G        TABELD  016322 G        TRE     040000 G        T$$PC = 000001          WRU     000400 G
RWU2    004000 G        TAP     040000 G        TYPLOC  011046          T$$PRO= 010010          X$ALWA= 000000
RWU3    010000 G        TBLPTR  002256 G        T$ARGC= 000003          T$$PTA= 010022          X$FALS= 000040
SAMSEC  002245 G        TD      000010 G        T$CODE= 003032          T$$RPT= 010007          X$OFFS= 000400
SAVCYL  002316 G        TDBCNT  002330 G        T$ERRN= 000002          T$$SOF= 010020          X$TRUE= 000020
SAVETD  026236          TDCASE  013732          T$EXCP= 000000          T$$SW = 010001          $PATCH  031250 G
SAVRPR  017542          TDCPY1  031430 G        T$FLAG= 000050          T$$TES= 010016

. ABS.  120114      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29952 WORDS  ( 117 PAGES)
DYNAMIC MEMORY AVAILABLE FOR  70 PAGES
CZRJKB.BIC,CZRJKB/C=[20,0]SVC34R.MLB,[20,12]CZRJKB.DOC,CZRJKB.HIS,CZRJKB
```

CROSS REFERENCE TABLE (CREF V04.00)

```
$$$ERR  13-185
$$$MFG   2-16     2-29     6-3      7-133    7-195    13-17    13-38    14-117   23-18    71-42    71-73    71-81    77 78
$PATCH  81-31#
A16     12-7#    16-33    16-38
A17     12-8#    16-33    16-38
ADR     11-57#
ANDMDR  14-107#  51-21
ANYWHR  13-18#   23-19    71-43*   71-51*   71-54    71-55    71-61    71-74    77-79
AOE     12-87#
ASSEMB   7-144    7-144
ATO     12-103#
AT1     12-104#
AT2     12-105#
AT3     12-106#
AT4     12-107#
AT5     12-108#
AT6     12-109#
AT7     12-110#
ATA     12-73#
BAI     12-28#
BEGBUF  34-94    81-63#
BELL    14-14    14-109   14-109   14-119   14-120
BGNPAT  13-8#    13-86    55-15
BIT0    11-57#
BIT00   11-57    11-57#
BIT01   11-57    11-57#
BIT02   11-57    11-57#
BIT03   11-57    11-57#
BIT04   11-57    11-57#
BIT05   11-57    11-57#
BIT06   11-57    11-57#
BIT07   11-57    11-57#
BIT08   11-57    11-57#
BIT09   11-57    11-57#
BIT1    11-57#
BIT10   11-57#
BIT11   11-57#   34-30    56-24
BIT12   11-57#   48-12    50-43    53-31    66-92
BIT13   11-57#   48-12    50-43    53-31    66-92
BIT14   11-57#   23-38    28-20    28-50    49-30    49-55    53-34    53-47    54-53    54-106   54-108   55-101   55-103
BIT15   11-57#   23-38    28-24    28-45    28-50    49-34    49-51    49-58    54-48    54-103   54-105   55-98    55-100
BIT2    11-57#
BIT3    11-57#
BIT4    11-57#
BIT5    11-57#
BIT6    11-57#
BIT7    11-57#
BIT8    11-57#   52-28    53-39    53-47    54-94    54-109   55-89    55-104   66-27    67-13
BIT9    11-57#   64-35    67-11
BOE     11-57#
BSE     12-181#  17-57
BUFINI  34-90    34-94#
BUSADR  13-49#   17-39    22-18*   33-33*   33-36*   42-22*   43-22*   43-35    43-37*   43-40*   49-37*   50-30*   51-30    54-21*
        55-21*
C$AU     7-144#  76-35
C$AUTO   7-144#  73-18
```

CROSS REFERENCE TABLE (CREF V04.00)

```
C$BRK     7-144#  77-106  78-14
C$BSEG    7 144#
C$BSUB    7-144#
C$CEFG    7 144#
C$CLCK    7-144#
C$CLEA    7-144#  74-62
C$CLOS    7-144#
C$CLP1    7-144#
C$CVEC    7-144#
C$DCLN    7-144#  72-10
C$DODU    7-144#
C$DRPT    7-144#
C$DU      7-144#  75-34
C$EDIT    7-144#   7-198
C$ERDF    7-144#  34-18   34-34
C$ERHR    7-144#  27-38   27-42   27-45   27-48   27 51   27-54   27-57   27-60   27-63   27-66   27-69   27-72   27-75
         27-78    27-81   27-84
C$ERRO    7-144#
C$ERSF    7-144#
C$ERSO    7-144#
C$ESCA    7-144#
C$ESEG    7-144#
C$ESUB    7-144#
C$ETST    7-144#  78-137
C$EXIT    7-144#  71-16   71-22   72-11   72-109  74-17   74-47
C$GETB    7-144#
C$GETW    7-144#
C$GMAN    7-144#  23-17   23-23   23-25   23-52   23-56   23-60   23-64   23-71   23-73   23-81   23-118  71-41   71-54
         71-61    71-70   71-78   71-80   71-85   71-87   71-92   71-98   71-103
C$GPHR    7-144#  35-21
C$GPLO    7-144#
C$GPRI    7-144#
C$INIT    7-144#  72-124
C$INLP    7-144#
C$MANI    7-144#  77-119
C$MEM     7-144#
C$MSG     7-144#  15-14   15-25   15-34   15-40   15-49
C$OPEN    7-144#
C$PNTB    7-144#  15-13   15-19   15-21   15-24   15-30   15-31   15-33   15-39   15 45   15-48   18-16   18-20   23-50
C$PNTF    7-144#  20-38   34-13   34-25   34-28   38-45   38-57   38-61   38-63   39-20   39-22   39-33   39-39   39-42
         39-46    42-38   47-38   51-43   51-48   51-51   51-55   52-21   52-24   52-41   52-45   52-48   52-53   71-37
         71-39    74-14   74-21   74-24   74-27   74-32   74-34   74-36
C$PNTS    7-144#
C$PNTX    7-144#  19-16   19-17   19-19   19-20   19-22   19-23   19-28   19-29
C$QIO     7-144#
C$RDBU    7-144#
C$REFG    7-144#  71-12   71-18   71-24   71-28
C$RESE    7-144#   7-144#  71-10
C$REVI    7-144#   7-198
C$RFLA    7-144#
C$RPT     7-144#  69-76
C$SEFG    7-144#
C$SPRI    7-144#  72-13   74-10
C$SVEC    7-144#
C$TPRI    7-144#
CALCTD   30-16#   33-22
```

CROSS REFERENCE TABLE (CREF V04.00)

CROSS REFERENCE TABLE (CREF V04.00)

J12

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

```
ILEV     12-63#   34-21
ILF      12-78#
ILR      12-79#
INPUTD   23-15#   77-121
INSERT   23-82    38-16#   54-82    55-78    64-20
INTEN    12-5#
INTLEV   13-55#   22-15    32-36    34-23*   36-15    41-18    54-61
IR       12-31#
ISR      11-57#
ITCNT    13-66#   26-21*   53-14*   53-21    54-15*   54-36    55-33*   55-51
IXE      11-57#
IXU      12-174#
J$JMP     7-144#  69-61    75-19    76-20
L$ACP     7-198#
L$APT     7-198#
L$AU     76-9#
L$AUT     7-198#
L$AUTO    7-198   73-10#
L$CCP     7-198#
L$CLEA    7-198   74-8#
L$CO      7-198#
L$DEPO    7-198#
L$DESC    7-198   14-27#
L$DESP    7-198#
L$DEVP    7-198#
L$DISP    7-198    8-8#
L$DLY     7-198#  24-11
L$DTP     7-198#
L$DTYP    7-198#
L$DU     75-8#
L$DUT     7-198#
L$DVTY    7-198   14-17#
L$EF      7-198#
L$ENVI    7-198#
L$ETP     7-198#
L$EXP1    7-198#
L$EXP4    7-198#
L$EXP5    7-198#
L$HARD    7-198   80-53    80-53#
L$HIME    7-198#
L$HPCP    7-198#
L$HPTP    7-198#
L$HW      7-198    9-10     9-10#
L$ICP     7-198#
L$INIT    7-198   71-8#
L$LADP    7-198#
L$LAST    7-198   81-84#   82-22
L$LOAD    7-198#
L$LUN     7-198#
L$MREV    7-198#
L$NAME    7-198#
L$PRIO    7-198#
L$PROT    7-198   70-8#
L$PRT     7-198#
L$REPP    7-198#
L$REV     7-198#
```

```
L$RPT    69-47#
L$SOFT   81-12    81 12#
L$SPC     7-198#
L$SPCP    7-198#
L$SPTP    7-198#
L$STA     7-198#
L$SW     10-10    10-10#
L$TEST    7-198#  71-33*   74-15
L$TIML    7-198#
L$UNIT    7-198#  35-46    72-8
L10000    9-10     9-25#
L10001   10-10    10-20#
L10002   15-14#
L10003   15-25#
L10004   15-34#
L10005   15-40#
L10006   15-49#
L10007   69-61    69-76#
L10011   71-16    71-22    72-11    72-109   72-124#
L10012   73-18#
L10013   74-17    74-47    74-62#
L10014   75-19    75-34#
L10015   76-20    76-35#
L10016   78-137#
L10017   80-53    80-62#
L10020   81-12    81-22#
L10021   82-16#
L10023   82-16    82-21#
LASLOC   13-51#   67-24*   78-6*    78-43    78-45
LBC      12-178#
LBT      12-68#
LCE      12-177#
LF       14-13    14-103   14-104   14-105   14-106   14-107   14-108   14-119   14-120
LINBUF   21-18    22-18    41-21    81-71#
LISHDR   13-33#   51-19    51-24    71-103
LIST     13-27#   34-60    39-15    47-29    72-44*   77-137
LISTDS   51-15#   77-139
LOADTD   66-14#   78-74
LOCYL    13-124#  65-13*   65-78*   65-79*   65-80    65-82*   65-85*   65-86    65-88*   65-89    65-101   65-104   65-107*  65-109
         65-129*
LODBSF   20-15#   23-85    54-87    55-82    64-26
LOE      11-57#
LOT      11-57#
LSCYL    13-44#   17-42    20-27    38-46    42-19    43-19    77-81
LSTRK    13-45#   20-30    38-49    42-21    43-21
MAKTRK   33-21#   51-27    77-127   78-90
MAPSEC   36-15#   37-15
MAXCYL   13-37#   61-31    61-33    71-44*   71-80    71-82    77-84*   77-148   78-38    78-111
MAXSEG   13-132#  29-29*   29-42    29-46    29-55*   29-65*
MAXTRK   13-42#   46-43    71-87
MAXVAL   61-12#   65-97
MCPE     12-10#
MERGE    32-14#   33-23
MESG1    80-55    80-67#
MESG13   14-113#  71-85
MESG14   14-114#  71-87
```

CROSS REFERENCE TABLE (CREF V04.00)

```
MESG15   14 1150  71-78
MESG16   14 1160  71-80
MESG17   14-1230  71-92
MESG18   14-1110  71-70
MESG19   14-1250  71-98
MESG22   14-1240  23-17
MESG23   14-1260  71-103
MESG38   23-23    23-1240
MESG39   23-25    23-1250
MESG4    80-57    80-680
MESG40   23 73    23-1260
MESG41   23-81    23-1270
MESG42   23-71    23-1280
MESG43   23-52    23-1290
MESG44   23-56    23-1300
MESG45   23-60    23-1310
MESG46   23-64    23-1320
MESG47   23-118   23-1330
MESG5    80-59    80-690
MESG6    80-61    80-700
MESG7    14-1120  71-41
MFGBUF   31-20    42-22    42-39    42-43    43-22    44-11    77-96    81-650
MINCYL   13-360   65-82    65-86    65-88    71-78    77-74
MINSEG   13-1310  29-28*   29-34    29-40    29-54*   29 64*
MINTRK   13-410   46-45    71-85    77-75
MODBY    13-320   20-18    23-17    23-34    38-53
MODTD    13-260   20-16    47-31    72-48*   77-141
MOM      12-1290
MOL      12-700   34-48
MOLINE   14-1000  34-50
MORETD   13-1330  23-118   72-66*   77-143
MPE      12-330
MSEG     12-2380  13-99    13-101   30-90    30-92
MSRDTD   14-930   50-21
MSWRTD   14-920   49-38
MTD      12-1550  49-54
MXF      12-340
NBA      12-1310
NED      12-370   34-15
NEGWRD   13-1250  17-31*   17-36    22-22*   33-37*   42-24*   43-25*   43-41*   49-40*   50-31*   54-22*   55-22*   55-38*
NEM      12-360
NEWCNT   13-570   53-42*   54-97*   55-92*   64-29*   78-4*    78-36    78-49    78-60    78-102*
NEWLST   52-150   78-70
NOGOOD   34-53    34-67    34-78    34-86    34-930
NOOP     12-2050
NORUN    34-19    34-35    34-46    34-920
NOWRCK   13-300   17-90    71-92
NULL     12-2390  23-92    28-48    29-50    30-29    30-33    30-43    30-50    30-96    30-112   30-149   30-163   30-164   30 165
         30-166   38-19    38-80    39-36    47-16    47-23    48-16    49-26    49-56    51-16    51-34    51-45    52-42    53-44
         53-45    53-46    66-95    66-96    66-97    66-98
O$APTS    7-1440   7-198
O$AU      7-1440   7-198
O$BGNR    7-1440   7-198
O$BGNS    7-1440   7-198
O$DU      7-1440   7-198
O$ERRT    7-1440   7-198
```

```
RMDATA   14 99#   33-32    54-20
RMEXT    13 138#  16-11*   16-20*   16-24*   16-30    35 32
RMTYPE   13-139#  16-12*   16-40*   19-25    35-30    40-21    56-21
RMR      12-80#
RPADR    13-136#  35-39*
RPAS     13-150#  24-44*
RPBA     13-145#  17-39*
RPBAE    13-163#
RPCC     13-158#
RPCS1    13-143#  17-47*   17-48    17-55    17-59*   17-80    25-14*   34-73*   35-19    40-15
RPCS2    13-147#  17-66    24-25*   24-26*   34-15    74-11*   74-12*
RPCS3    13-164#
RPDA     13-146#  17-38*
RPDB     13-152#
RPDC     13-157#  17-37*
RPDS     13-148#  17-26    17-32    17-75    17 85    24-27    24-29*   24-32    34-14    34 21    34-38    34-48    34-62    34-74
RPDT     13-154#  34-29
RPEC1    13-161#
RPEC2    13-162#
RPER1    13-149#
RPER2    13-159#
RPER3    13-160#  17-57
RPLA     13-151#
RPMR1    13-153#  17-41*   17-46*
RPOF     13-156#  17-40*
RPSN     13-155#
RPVEC    13-137#  35-40*   35-41*
RPWC     13-144#  17-36*   17-60
RTYCNT   13-70#   26-26*   26-28    26-33*   33-43    34-55    34-69    34-80    34-88    42-29    43-30    43-46    49-45    50-37
         54-28    54-42*   55-27    55-43    55-61*   77-68*
RWU1     12-188#
RWU2     12-189#
RWU3     12-190#
S$LSYM    7-144#   9-25#   10-20#   15-14#   15-25#   15-34#   15-40#   15-49#   23-17    23-17    23-17    23-17#   23-23    23-23
         23-23    23-23#   23-25    23-25    23-25    23-25#   23-52    23-52    23-52    23-52#   23-56    23-56    23-56    23-56#
         23-60    23-60    23-60    23-60#   23-64    23-64    23-64    23-64#   23-71    23-71    23-71    23-71#   23-73    23-73
         23-73    23-73#   23-81    23-81    23-81    23-81#   23-118   23-118   23-118   23-118#  69-76#   71-41    71-41    71-41
         71-41#   71-54    71-54    71-54    71-54#   71-61    71-61    71-61    71-61#   71-70    71-70    71-70    71-70#   71-78
         71-78    71-78    71-78#   71-80    71-80    71-80    71-80#   71-85    71-85    71-85    71-85#   71-87    71-87    71-87
         71-87#   71-92    71-92    71-92    71-92#   71-98    71-98    71-98    71-98#   71-103   71-103   71-103   71-103#  72 124#
         73-18#   74-62#   75-34#   76-35#   78-137#  80-62#   81-22#
SAMSEC   13-56#   30-55*   30-85    30-102   30-143*
SAVCYL   13-80#   78-15*   78-51
SAVETD   23-97    66-42#   78-76
SAVRPR   24-40    26-23    40-14#
SAVTRK   13-81#   72-24    78-16*   78-52
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     77-108   78-3#    78-113
```

```
SCANIT  13-54#   52-16*   52-18    77-69*
SCANR   13 25#   38-51    50-57    72-40*   77-107
SCF     12-169#
SDF     12-172#
SEARCH  12-215#
SEC50   13-83#   21-20    32-55    32-60    41-22    43-49    56 50
SECADD  13-127#  23-71    36-19    36-25*   37-14    37-20*   37-26*   38-40    54-65    54-67*   54-69*   54-70*   54-86    56-47*
        56-48*   56-50*   56-53*   60-21*   60-24*   64-25
SEEK    12-206#
SEIZE   24-25#   25-13
SFPTBL  10-10#
SIZE70  16-11#   35-29
SKI     12-180#
SOFSW   13-65#   34-93*   77-86
SORT    43-16    44-10#
SPIRAL  46-42#   77-147   78-35    78-53
SUPRSS  13-61#   23-103*  23-107*  23-115*  24-38    24-39*   24-42*   27-16    42-17*   42-54*   53-38*   54-93*   55-88*   57-40*
        78-3*    78-31*
SVCGBL   7-144#   7-153#   7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    8-8      8-8      8-8      9-10     9-10     9-10     9-10     9-10     9-10    10-10
        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    69-47    69-47
        69-47    70-8     70-8     70-8     71-8     71-8     71-8     73-10    73-10    73-10    74-8     74-8     74-8     75-8
        75-8     75-8     76-9     76-9     76-9     80-53    80-53    80-53    81-12    81-12    81-12    81-84    81-84    81-84
        81-84#
SVCINS   7-144#   7-150#   7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
         7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    8-8      8-8
         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    19-16    19-16
        19-16    19-16    19-16    19-16    19-16    19-16    19-16    19-16    19-17    19-17    19-17    19-17    19-17    19-17
        19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17
        19-17    19-17    19-17    19-17    19-19    19-19    19-19    19-19    19-19    19-19    19-19    19-19    19-19    19 19
        19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20
```

19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-22	19-22	19-22	19-22
19 22	19-22	19-22	19-22	19-22	19-22	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23
19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23
19 23	19-23	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-28	19-29	19-29
19-29	19-29	19-29	19-29	19-29	19-29	19-29	19-29	19-29	19-29	19-29	19-29	20-38	20-38
20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	20-38	23-17	23-17	23-17	23-17
23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-23	23-23
23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23	23-23
23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25	23-25
23-25	23-25	23-50	23-50	23-50	23-50	23-50	23-50	23-50	23-50	23-50	23-50	23-52	23-52
23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52
23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56
23-56	23-56	23-60	23-60	23-60	23-60	23-60	23-60	23-60	23-60	23-60	23-60	23-60	23-60
23-60	23-60	23-60	23-60	23-64	23-64	23-64	23-64	23-64	23-64	23-64	23-64	23-64	23-64
23-64	23-64	23-64	23-64	23-64	23-64	23-71	23-71	23-71	23-71	23-71	23-71	23-71	23-71
23-71	23-71	23-71	23-71	23-71	23-71	23-71	23-71	23-73	23-73	23-73	23-73	23-73	23-73
23-73	23-73	23-73	23-73	23-73	23-73	23-73	23-73	23-73	23-73	23-81	23-81	23-81	23-81
23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-118	23-118
23-118	23-118	23-118	23-118	23-118	23-118	23-118	23-118	23-118	23-118	24-11	24-11	24-11	24-11
24-11	24-11	24-11	24-11	24-11	24-11	24-11	24-11	24-11	24-11	24-11	24-11	27-38	27-38
27-38	27-38	27-38	27-38	27-38	27-38	27-42	27-42	27-42	27-42	27-42	27-42	27-42	27-42
27-45	27-45	27-45	27-45	27-45	27-45	27-45	27-45	27-48	27-48	27-48	27-48	27-48	27-48
27-48	27-48	27-51	27-51	27-51	27-51	27-51	27-51	27-51	27-51	27-54	27-54	27-54	27-54
27-54	27-54	27-54	27-54	27-57	27-57	27-57	27-57	27-57	27-57	27-57	27-57	27-60	27-60
27-60	27-60	27-60	27-60	27-60	27-60	27-63	27-63	27-63	27-63	27-63	27-63	27-63	27-63
27-66	27-66	27-66	27-66	27-66	27-66	27-66	27-66	27-69	27-69	27-69	27-69	27-69	27-69
27-69	27-69	27-72	27-72	27-72	27-72	27-72	27-72	27-72	27-72	27-75	27-75	27-75	27-75
27-75	27-75	27-75	27-75	27-78	27-78	27-78	27-78	27-78	27-78	27-78	27-78	27-81	27-81
27-81	27-81	27-81	27-81	27-81	27-81	27-84	27-84	27-84	27-84	27-84	27-84	27-84	27-84
34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-13	34-18	34-18
34-18	34-18	34-18	34-18	34-18	34-18	34-25	34-25	34-25	34-25	34-25	34-25	34-25	34-25
34-25	34-25	34-28	34-28	34-28	34-28	34-28	34-28	34-28	34-28	34-28	34-28	34-34	34-34
34-34	34-34	34-34	34-34	34-34	34-34	35-21	35-21	35-21	35-21	35-21	35-21	35-22	35-22
38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45	38-45
38-45	38-45	38-57	38-57	38-57	38-57	38-57	38-57	38-57	38-57	38-57	38-57	38-61	38-61
38-61	38-61	38-61	38-61	38-61	38-61	38-61	38-61	38-61	38-61	38-63	38-63	38-63	38-63
38-63	38-63	38-63	38-63	38-63	38-63	39-20	39-20	39-20	39-20	39-20	39-20	39-20	39-20
39-20	39-20	39-20	39-20	39-22	39-22	39-22	39-22	39-22	39-22	39-22	39-22	39-22	39-22
39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33	39-33
39-33	39-33	39-33	39-33	39-33	39-33	39-39	39-39	39-39	39-39	39-39	39-39	39-39	39-39
39-39	39-39	39-39	39-39	39-42	39-42	39-42	39-42	39-42	39-42	39-42	39-42	39-42	39-42
39-42	39-42	39-46	39-46	39-46	39-46	39-46	39-46	39-46	39-46	39-46	39-46	42-38	42-38
42-38	42-38	42-38	42-38	42-38	42-38	42-38	42-38	42-38	42-38	47-38	47-38	47-38	47-38
47-38	47-38	47-38	47-38	47-38	47-38	47-38	47-38	47-38	47-38	47-38	47-38	51-43	51-43
51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-43	51-48	51-48
51-48	51-48	51-48	51-48	51-48	51-48	51-48	51-48	51-48	51-48	51-51	51-51	51-51	51-51
51-51	51-51	51-51	51-51	51-51	51-51	51-51	51-51	51-55	51-55	51-55	51-55	51-55	51-55
51-55	51-55	51-55	51-55	52-21	52-21	52-21	52-21	52-21	52-21	52-21	52-21	52-21	52-21
52-21	52-21	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-24	52-41	52-41
52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41	52-41
52-41	52-41	52-41	52-41	52-45	52-45	52-45	52-45	52-45	52-45	52-45	52-45	52-45	52-45
52-45	52-45	52-48	52-48	52-48	52-48	52-48	52-48	52-48	52-48	52-48	52-48	52-48	52-48
52-53	52-53	52-53	52-53	52-53	52-53	52-53	52-53	52-53	52-53	69-61	69-61	69-61	69-61
69-76	69-76	71-10	71-10	71-12	71-12	71-12	71-12	71-14	71-14	71-16	71-16	71-16	71-16
71-18	71-18	71-18	71-18	71-20	71-20	71-22	71-22	71-22	71-22	71-24	71-24	71-24	71-24
71-26	71-26	71-28	71-28	71-28	71-28	71-30	71-30	71-37	71-37	71-37	71-37	71-37	71-37

```
        71-37   71-37   71-37   71-37   71-39   71-39   71-39   71-39   71-39   71-39   71-39   71-39   71-39   71-39
        71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41   71-41
        71-41   71-41   71-54   71-54   71-54   71-54   71-54   71-54   71-54   71-54   71-54   71-54   71-54   71-54
        71 61   71-61   71-61   71-61   71-61   71-61   71-61   71-61   71-61   71-61   71-61   71-61   71-70   71-70
        71-70   71-70   71-70   71-70   71-70   71-70   71-70   71-70   71-70   71-70   71-78   71-78   71-78   71-78
        71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-78   71-80   71-80
        71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80   71-80
        71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85   71-85
        71-85   71-85   71-87   71-87   71-87   71-87   71-87   71-87   71-87   71-87   71-87   71-87   71-87   71-87
        71-87   71-87   71-87   71-87   71-92   71-92   71-92   71-92   71-92   71-92   71-92   71-92   71-92   71-92
        71-92   71-92   71-98   71-98   71-98   71-98   71-98   71-98   71-98   71-98   71-98   71-98   71-98   71-98
        71-103  71-103  71-103  71-103  71-103  71-103  71-103  71-103  71-103  71-103  71-103  71-103  72-10   72-10
        72-11   72-11   72-11   72-11   72-13   72-13   72-13   72-13   72-109  72-109  72-109  72-109  72-124  72-124
        73-18   73-18   74-10   74-10   74-10   74-10   74-14   74-14   74-14   74-14   74-14   74-14   74-14   74-14
        74-14   74-14   74-17   74-17   74-17   74-17   74-21   74-21   74-21   74-21   74-21   74-21   74-21   74-21
        74-21   74-21   74-21   74-21   74-21   74-21   74-24   74-24   74-24   74-24   74-24   74-24   74-24   74-24
        74-24   74-24   74-24   74-24   74-24   74-24   74-27   74-27   74-27   74-27   74-27   74-27   74-27   74-27
        74-27   74-27   74-27   74-27   74-27   74-27   74-32   74-32   74-32   74-32   74-32   74-32   74-32   74-32
        74-32   74-32   74-32   74-32   74-32   74-32   74-34   74-34   74-34   74-34   74-34   74-34   74-34   74-34
        74-34   74-34   74-34   74-34   74-36   74-36   74-36   74-36   74-36   74-36   74-36   74-36   74-36   74-36
        74-36   74-36   74-36   74-36   74-47   74-47   74-47   74-47   74-62   74-62   75-19   75-19   75-19   75-19
        75-34   75-34   76-20   76-20   76-20   76-20   76-35   76-35   77-106  77-106  77-119  77-119  77-120  77-120
        78-14   78-14   78-137  78-137  80-53   80-53   80-55   80-55   80-55   80-55   80-55   80-55   80-55   80-55
        80-57   80-57   80-57   80-57   80-57   80-57   80-57   80-57   80-59   80-59   80-59   80-59   80-59   80-59
        80-59   80-59   80-59   80-59   80-61   80-61   80-61   80-61   80-61   80-61   80-61   80-61   80-61   80-61
        80-62   80-62   81-12   81-12   81-22   81-22   81-84   81-84   81-84   81-84   81-84   81-84   82-16   82-16
        82-16   82-16
SVCSUB   7-144#   7-152#
SVCTAG   7-144#   7-154#   9-25    9-25    9-25    10-20   10-20   10-20   15-14   15-14   15-14   15-25   15-25   15-25
        15-34   15-34   15-34   15-40   15-40   15-40   15-49   15-49   15-49   23-17   23-17   23-17   23-23   23-23
        23-23   23-25   23-25   23-25   23-52   23-52   23-52   23-56   23-56   23-56   23-60   23-60   23-60   23-64
        23-64   23-64   23-71   23-71   23-71   23-73   23-73   23-73   23-81   23-81   23-81   23-118  23-118  23-118
        69-76   69-76   69-76   71-41   71-41   71-41   71-54   71-54   71-54   71-61   71-61   71-61   71-70   71-70
        71-70   71-78   71-78   71-78   71-80   71-80   71-80   71-85   71-85   71-85   71-87   71-87   71-87   71-92
        71-92   71-92   71-98   71-98   71-98   71-103  71-103  71-103  72-124  72-124  72-124  73-18   73-18   73-18
        74-62   74-62   74-62   75-34   75-34   75-34   76-35   76-35   76-35   78-137  78-137  78-137  80-62   80-62
        80-62   81-22   81-22   81-22   82-16   82-16   82-16   82-21   82-21   82-21
SVCTST   7-144#   7-151#   77-66   77-66   77-66
T$$AU   76-9#   76-20   76-35
T$$AUT   73-10#   73-18
T$$CLE   74-8#   74-17   74-47   74-62
T$$DAT   82-16   82-16#   82-21
T$$DU   75-8#   75-19   75-34
T$$HAR   80-53   80-53#   80-62
T$$HW   9-10   9-10#   9-25
T$$INI   71-8#   71-16   71-22   72-11   72-109   72-124
T$$MSG   15-9#   15-14   15-16#   15-25   15-27#   15-34   15-36#   15-40   15-42#   15-49
T$$PC   82-15#   82-22
T$$PRO   70-8#
T$$PTA   82-15#   82-16   82-16#
T$$RPT   69-47#   69-61   69-76
T$$SOF   81-12   81-12#   81-22
T$$SW   10-10   10-10#   10-20
T$$TES   77-66#   78-137
T$ARGC   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198#   7-198#   7 198#
        7-198#   7-198#   7-198#   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#  15-24   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#  15-31#  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#  19-16   19-16   19-16#  19-17   19-17   19-17   19-17   19-17   19-17
                19-17   19-17   19-17   19-17#  19-17#  19-17#  19-17#  19-17#  19-17#  19-17#  19-17#  19-19   19-19   19-19#
                19-20   19-20   19-20   19-20   19-20   19-20   19-20   19-20   19-20   19-20#  19-20#  19-20#  19-20#  19-20#
                19-20#  19-20#  19-20#  19-22   19-22   19-22#  19-23   19-23   19-23   19-23   19-23   19-23   19-23   19-23
                19-23   19-23#  19-23#  19-23#  19-23#  19-23#  19-23#  19-23#  19-23#  19-28   19-28   19-28#  19-29   19-29
                19-29   19-29   19-29#  19-29#  19-29#  20-38   20-38   20-38   20-38#  20-38#  23-50   23-50   23-50#  34-13
                34-13   34-13   34-13#  34-13#  34-25   34-25   34-25#  34-28   34-28   34-28#  38-45   38-45   38-45   38-45
                38-45   38-45#  38-45#  38-45#  38-45#  38-57   38-57   38-57#  38-61   38-61   38-61   38-61#  38-61#  38-63
                38-63   38-63#  39-20   39-20   39-20   39-20#  39-20#  39-22   39-22   39-22#  39-33   39-33   39-33   39-33
                39-33   39-33   39-33   39-33#  39-33#  39-33#  39-33#  39-33#  39-33#  39-39   39-39   39-39   39-39#  39-39#
                39-42   39-42   39-42   39-42#  39-42#  39-46   39-46   39-46#  42-38   42-38   42-38   42-38#  42-38#  47-38
                47-38   47-38   47-38   47-38   47-38#  47-38#  47-38#  47-38#  47-38#  51-43   51-43   51-43   51-43   51-43#
                51-43#  51-48   51-48   51-48   51-48#  51-48#  51-51   51-51   51-51   51-51#  51-51#  51-55   51-55   51-55#
                52-21   52-21   52-21   52-21#  52-21#  52-24   52-24   52-24#  52-41   52-41   52-41   52-41   52-41   52-41
                52-41   52-41#  52-41#  52-41#  52-41#  52-41#  52-41#  52-45   52-45   52-45   52-45#  52-45#  52-48   52-48
                52-48   52-48#  52-48#  52-53   52-53   52-53#  71-37   71-37   71-37#  71-39   71-39   71-39#  74-14   74-14
                74-14#  74-21   74-21   74-21   74-21   74-21#  74-21#  74-21#  74-21#  74-24   74-24   74-24   74-24   74-24#
                74-24#  74-27   74-27   74-27   74-27   74-27#  74-27#  74-27#  74-27#  74-32   74-32   74-32   74-32   74-32#
                74-32#  74-34   74-34   74-34   74-34#  74-34#  74-36   74-36   74-36   74-36   74-36#  74-36#  74-36#
T$CODE          23-17   23-17   23-17   23-17#  23-17#  23-17#  23-23   23-23   23-23   23-23#  23-23#  23-23#  23-25   23-25
                23-25   23-25#  23-25#  23-25#  23-52   23-52   23-52   23-52#  23-52#  23-52#  23-56   23-56   23-56   23-56#
                23-56#  23-56#  23-60   23-60   23-60   23-60#  23-60#  23-60#  23-60#  23-64   23-64   23-64#  23-64#  23-64#
                23-71   23-71   23-71   23-71#  23-71#  23-71#  23-73   23-73   23-73   23-73#  23-73#  23-73#  23-81   23-81
                23-81   23-81#  23-81#  23-81#  23-118  23-118  23-118  23-118#  23-118#  23-118#  71-41   71-41   71-41   71-41#
                71-41#  71-41#  71-54   71-54   71-54   71-54#  71-54#  71-54#  71-54#  71-61   71-61   71-61   71-61#  71-61#
                71-70   71-70   71-70   71-70#  71-70#  71-70#  71-78   71-78   71-78   71-78#  71-78#  71-78#  71-80   71-80
                71-80   71-80#  71-80#  71-80#  71-85   71-85   71-85   71-85#  71-85#  71-85#  71-87   71-87   71-87   71-87#
                71-87#  71-87#  71-92   71-92   71-92   71-92#  71-92#  71-92#  71-98   71-98   71-98   71-98#  71-98#  71-98#
                71-103  71-103  71-103  71-103#  71-103#  71-103#  80-55   80-55   80-55   80-55#  80-55#  80-55#  80-57   80-57
                80-57   80-57#  80-57#  80-57#  80-59   80-59   80-59   80-59#  80-59#  80-59#  80-61   80-61   80-61   80-61#
                80-61#  80-61#
T$ERRN           7-144#  27-38   27-38#  27-42   27-42#  27-45   27-45#  27-48   27-48#  27-51   27-51#  27-54   27-54#  27-57
                27-57#  27-60   27-60#  27-63   27-63#  27-66   27-66#  27-69   27-69#  27-72   27-72#  27-75   27-75#  27-78
                27-78#  27-81   27-81#  27-84   27-84#  34-18   34-18#  34-34   34-34#
T$EXCP          23-17   23-17#  23-23   23-23#  23-25   23-25#  23-52   23-52#  23-56   23-56#  23-60   23-60#  23-64   23-64#
                23-71   23-71#  23-73   23-73#  23-81   23-81#  71-41   71-41#  71-78   71-78#  71-80   71-80#  71-85   71-85#
                71-87   71-87#  80-55   80-55#  80-57   80-57#  80-59   80-59#  80-61   80-61#
T$FLAG          69-61   69-61#  69-61#  71-16   71-16   71-16#  71-16#  71-22   71-22   71-22#  71-22#  72-11   72-11   72-11#
                72-11#  72-109  72-109  72-109#  72-109#  74-17   74-17   74-17#  74-17#  74-47   74-47   74-47#  74-47#  75-19
                75-19#  75-19#  76-20   76-20#  76-20#
T$FREE          81-84   82-22#
T$GMAN           7-144#  23-17#  23-17#  23-23#  23-23#  23-25#  23-25#  23-52#  23-52#  23-56#  23-56#  23-60#  23-60#  23-64#
                23-64#  23-71#  23-71#  23-73#  23-73#  23-81#  23-81#  71-41#  71-41#  71-78#  71-78#  71-80#  71-80#  71-85#
                71-85#  71-87#  71-87#
T$HILI          23-17   23-17#  23-23   23-23#  23-25   23-25#  23-52   23-52#  23-56   23-56#  23-60   23-60#  23-64   23-64#
                23-71   23-71#  23-73   23-73#  23-81   23-81#  71-41   71-41#  71-78   71-78#  71-80   71-80#  71-85   71-85#
                71-87   71-87#  80-55   80-55#  80-57   80-57#  80-59   80-59#  80-61   80-61#
T$LAST           7-144#  81-84#  82-15
T$LOLI          23-17   23-17#  23-23   23-23#  23-25   23-25#  23-52   23-52#  23-56   23-56#  23-60   23-60#  23-64   23-64#
                23-71   23-71#  23-73   23-73#  23-81   23-81#  71-41   71-41#  71-78   71-78#  71-80   71-80#  71 85   71-85#
                71-87   71-87#  80-55   80-55#  80-57   80-57#  80-59   80-59#  80-61   80-61#
T$LSYM           7-144   7-144#   9-25   10-20   15-14   15-25   15-34   15-40   15-49   69-76   72-124  73-18   74-62   75-34
```

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

	71-80	71-85	71-87	71-92	71-98	71-103	72 10	72-11	72-13	72-109	72-124	73-18	74-10	74-14
	74 17	74-21	74-24	74-27	74-32	74 34	74-36	74-47	74-62	75-34	76-35	77-106	77-119	78-14
	78 137													
T$TSTS	7-144#	77-66#												
T1	8-8	77-66#												
TABELD	35-17#	72-7												
TAP	12-130#													
TBLPTR	13-63#	46-16	46-22*	78-43	78-106*									
TD	12-229#	13-88	13-114	13-115	28-17	28-26	28-30	37-17	54-50	60-19				
TDBCNT	13-87#	49-40	50-23	50-31										
TDCASE	28-16#	30-49												
TDCPY1	15-19	30-21	50-42*	50-43*	50-44*	50-47	81-48#							
TDCPY2	15-19	50-45*	50-46*	50-49	81-48#									
TDCPY3	23-39*	23-52	23-53	23-65	30-28	30-48	81-48#							
TDCPY4	23-45*	23-56	23-57	23-66	81-48#									
TDCPY5	23-45*	23-60	23-61	23-67	81-48#									
TDCPY6	23-45*	23-64	23-68	81-48#										
TDDEF	14-106#	51-18												
TDDUMP	23-33	23-112	39-14#	51-22										
TDLMTS	13-113#	23-75	23-77	28-16	49-23	55-74	55-76							
TDSBLK	30-17	30-20	32-15	32-23	33-33	41-23	54-18	55-18	57-18	81-70#				
TDSCAN	53-14#	78-18												
TDTMP1	66-44	66-66	81-58#											
TDTMP2	81-58#													
TDTMP3	81-58#													
TDTMP4	81-58#													
TDTMP5	81-58#													
TDTMP6	81-58#													
TDVALU	13-88#													
TDWRD1	15-21	15-31	23-38*	28-20	28-24	28-45	28-50	30-22	39-29	47-33	48-10	49-30	49-34	49-37
	49-51*	49-55*	49-58	50-22	50-30	50-41	54-48	54-53	54-99	54-103	54-106	55-94	55-98	55-101
	64-31	66-15	66-43	66-67	66-90	81-38#								
TDWRD2	15-21	15-31	30-23	39-23	47-35	81-38#								
TDWRD3	15-31	23-65*	23-88	30-24	38-18	38-26	38-29	39-34	47-16	47-18	47-21	49-24	51-16	60-14
	81-38#													
TDWRD4	15-31	23-66*	30-25	81-38#										
TDWRD5	15-31	23-67*	30-26	81-38#										
TDWRD6	15-31	23-68*	30-27	81-38#										
TEMP1	13-121#	59-11	59-13	59-21*	65-55*	65-61*	65-64	65-100						
TEMP2	13-122#	59-11	59-13	59-16*	59-18*	59-21*	59-37*	59-39*	59-41	61-26	62-44	63-18	65-56*	65-62*
	65-64													
TEMPA	13-60#	29-18*	29-57	65-35*	65-58									
TEXT	13-118#	23-32*	23-111*	39-20	51-18*	51-21*								
TKWCNT	13-120#	22-22	32-20	33-37	51-31	54-22								
TMDLTA	30-57*	30-122	30-129*	30-172#										
TMPBSF	20-27	20-30	20-32	20-33	23-83	31-27	31-29	32-39*	32-40*	38-25*	38-39*	38-40*	38-41*	38-46
	38-49	54-84	54-86*	55-80	64-22	64-25*	81-64#							
TMPCYL	13-39#	71-44	71-82*											
TOLER	54-76	55-68	57-17#											
TOLRAN	59-11#	59-40	65-63											
TRE	12-11#	17-55												
TYPLOC	15-10	15-17	15-28	15-37	15-43	18-15#								
UAM	11-57#													
UNIT	13-135#	35-21	35-45*	35-46	72-5*	72-6*	72-8							
UNS	12-92#													
UPDSCR	64-14#	65-106												

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

CROSS REFERENCE TABLE (CREF V04.00)

```
ENDSW    1-614#  7-144#  10-20
ENDTST   1-624#  7-144#  78-137
EQUALS   1-642#  7-144#  11-57
ERRDF    1-714#  7-144#  34-18   34-34
ERRHRD   1-718#  7-144#  27-38   27-42   27-45   27 48   27-51   27-54   27-57   27-60   27-63   27-66   27-69   27-72
         27-75   27-78   27-81   27-84
ERROR    1-722#  7-144#
ERRSF    1-726#  7-144#
ERRSOF   1-730#  7-144#
ERRTBL   1-734#  7-144#
ESCAPE   1-744#  7-144#
EXIT     1-771#  7-144#  69-61   71-16   71-22   72-11   72-109  74-17   74-47   75-19   76-20
FEQUAL   1-810#  7-144#
GETBYT   1-824#  7-144#
GETPRI   1-834#  7-144#
GETWOR   1-829#  7-144#
GMANIA   1-839#  7-144#
GMANID   1-848#  7-144#  23-17   23-23   23-25   23-52   23-56   23-60   23-64   23-71   23-73   23-81   71-41   71-78
         71-80   71-85   71-87
GMANIL   1-859#  7-144#  23-118  71-54   71-61   71-70   71-92   71-98   71-103
GPHARD   1-868#  7-144#  35-21
GPRMA    1-874#  7-144#  80-55   80-57
GPRMD    1-903#  7-144#  23-17   23-17#  23-23   23-23#  23-25   23-25#  23-52   23-52#  23-56   23-56#  23-60   23-60#
         23-64   23-64#  23-71   23-71#  23-73   23-73#  23-81   23-81#  71-41   71-41#  71-78   71-78#  71-80   71-80#
         71-85   71-85#  71-87   71-87#  80-59   80-61
GPRML    1-934#  7-144#  23-118  23-118#  71-54   71-54#  71-61   71-61#  71-70   71-70#  71-92   71-92#  71-98   71-98#
         71-103  71-103#
HEADER   1-954#  7-144#   7-198
INLOOP   1-962#  7-144#
IOSETU   1-966#  7-144#
IOSTAR   1-974#  7-144#
KT11     1-982#  7-144#
LASTAD   1-;47#  7-144#  81-84
M$BYTE   1-D00#  7-144#   7-198   7-198   7-198   7-198#
M$CHEC   1-E18#  7-144#  69-61   69-61#  71-16   71-16#  71-22   71-22#  72-11   72-11#  72-109  72-109#  74-17   74-17#
         74-47   74-47#  75-19   75-19#  76-20   76-20#
M$CNTO   1-E82#  7-144#  23-17   23-17#  23-23   23-23#  23-25   23-25#  23-52   23-52#  23-56   23-56#  23-60   23-60#
         23-64   23-64#  23-71   23-71#  23-73   23-73#  23-81   23-81#  23-118  23-118#  71-41   71-41#  71-54   71-54#
         71-61   71-61#  71-70   71-70#  71-78   71-78#  71-80   71-80#  71-85   71-85#  71-87   71-87#  71-92   71-92#
         71-98   71-98#  71-103  71-103#  80-55   80-55#  80-57   80-57#  80-59   80-59#  80-61   80-61#
M$COUN   1-D66#  7-144#  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#  15-33   15-33#  15-39   15-39#
         15-45   15-45   15-45#  15-48   15-48#  18-16   18-16#  18-20   18-20   18-20#  19-16   19-16#  19-17   19-17
         19-17   19-17   19-17   19-17   19-17   19-17#  19-19   19-19#  19-20   19-20   19-20   19-20   19-20   19-20
         19-20   19-20#  19-22   19-22#  19-23   19-23   19-23   19-23   19-23   19-23   19-23   19-23#  19-28   19-28#
         19-29   19-29   19-29#  20-38   20-38#  23-50   23-50#  34-13   34-13#  34-25   34-25#  34-28   34-28#  38-45
         38-45   38-45   38-45#  38-57   38-57#  38-61   38-61#  38-63   38-63#  39-20   39-20#  39-22   39-22#  39-33
         39-33   39-33   39-33   39-33   39-33#  39-39   39-39#  39-42   39-42#  39-46   39-46#  42-38   42-38#  47-38
         47-38   47-38   47-38#  51-43   51-43   51-43#  51-48   51-48#  51-51   51-51#  51-55   51-55#  52-21   52-21#
         52-24   52-24#  52-41   52-41   52-41   52-41   52-41   52-41#  52-45   52-45#  52-48   52-48#  52-53   52-53#
         71-37   71-37#  71-39   71-39#  74-14   74-14#  74-21   74-21   74-21#  74-24   74-24   74-24#  74-27   74-27
         74-27#  74-32   74-32   74-32#  74-34   74-34#  74-36   74-36   74-36#
M$DATA   1-B67#  7-144#   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198
          7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198
          7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198   7-198#  7-198#  14-17
         14-17#  14-27   14-27#
```

```
M$DECR    1 D29#   7-144#   9-25     9-25#    10-20    10-20#   10-21    10-21#   15-14    15-14#   15-25    15-25#   15-34    15-34#
         15-40    15-40#   15-49    15-49#   68-1     68-1#    69-76    69-76#   70-12    70-12#   72-124   72-124#  73-18    73-18#
         74-62    74-62#   75-34    75-34#   76-35    76-35#   76-36    76-36#   78-137   78-137#  79-1     79-1#    80-62    80-62#
         81 22    81-22#   81-85    81-85#   82-16    82-16#
M$DEFA    1-E70#   7-144#  23-17    23-17#   23-23    23-23#   23-25    23-25#   23-52    23-52#   23-56    23-56#   23-60    23-60#
         23-64    23-64#   23-71    23-71#   23-73    23-73#   23-81    23-81#   23-118   23-118#  71-41    71-41#   71-54    71-54#
         71-61    71-61#   71-70    71-70#   71-78    71-78#   71-80    71-80#   71-85    71-85#   71-87    71-87#   71-92    71-92#
         71-98    71-98#   71-103   71-103#  80-55    80-55#   80-57    80-57#   80-59    80-59#   80-61    80-61#
M$ENDE    1-D74#   7-144#   9-25#    10-20#   10-21#   15-14#   15-25#   15-34#   15-40#   15-49#   68-1#    69-76#   72-124#  73-18#
         74-62#   75-34#   76-35#   76-36#   78-137#  79-1#    80-62#   81-22#   81-85#
M$ERRI    1-B49#   7-144#  27-38    27-38#   27-42    27-42#   27-45    27-45#   27-48    27-48#   27-51    27-51#   27-54    27-54#
         27-57    27-57#   27-60    27-60#   27-63    27-63#   27-66    27-66#   27-69    27-69#   27-72    27-72#   27-75    27-75#
         27-78    27-78#   27-81    27-81#   27-84    27-84#   34-18    34-18#   34-34    34-34#
M$ESCA    1-D06#   7-144#
M$ESCS    1-D10#   7-144#
M$EXCP    1 E01#   7-144#  23-17    23-17    23-17#   23-23    23-23    23-23#   23-25    23-25    23-25#   23-52    23-52    23-52#
         23-56    23-56    23-56#   23-60    23-60    23-60#   23-64    23-64#   23-71    23-71    23-71#   23-73    23-73    23-73#
         23-73#   23-81    23-81    23-81#   71-41    71-41    71-41#   71-78    71-78    71-78#   71-80    71-80#   71-85    71-85
         71-85    71-85#   71-87    71-87    71-87#   80-55    80-55    80-55#   80-57    80-57    80-57#   80-59    80-59    80-59#
         80-61    80-61    80-61#
M$EXIT    1-D14#   7-144#  69-61#   71-16    71-16#   71-22    71-22#   72-11    72-11#   72-109   72-109#  74-17    74-17#   74-47
         74-47#   75-19#   76-20#
M$EXSE    1-D22#   7-144#  69-61#   71-16#   71-22#   72-11#   72-109#  74-17#   74-47#   75-19#   76-20#
M$EXTJ    1-D18#   7-144#  69-61    69-61#   71-16#   71-22#   72-11#   72-109#  74-17#   74-47#   75-19    75-19#   76-20    76-20#
M$GEN     1-D38#   7-144#   7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
          7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
          7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198#   7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   8-8      8-8#     9-10     9-10     9-10#    9-10#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   8-8      8-8#     9-10     9-10     9-10#    9-10#
          9-25     9-25#   10-10    10-10    10-10#   10-10#   10-20    10-20#   14-17    14-17#   14-27    14-27#   15-9     15-9#
         15-14    15-14#   15-16    15-16#   15-25    15-25#   15-27    15-27#   15-34    15-34#   15-36    15-36#   15-40    15-40#
         15-42    15-42#   15-49    15-49#   23-17    23-17#   23-23    23-23#   23-25    23-25#   23-52    23-52#   23-56    23-56#
         23-60    23-60#   23-64    23-64#   23-71    23-71#   23-73    23-73#   23-81    23-81#   23-118   23-118#  69-47    69-47#
         69-76    69-76#   70-8     70-8#    71-8     71-8#    71-41    71-41#   71-54    71-54#   71-61    71-61#   71-70    71-70#
         71-78    71-78#   71-80    71-80#   71-85    71-85#   71-87    71-87#   71-92    71-92#   71-98    71-98#   71-103   71-103#
         72-124   72-124#  73-10    73-10#   73-18    73-18#   74-8     74-8#    74-62    74-62#   75-8     75-8#    75-34    75-34#
         76-9     76-9#    76-35    76-35#   77-66    77-66#   78-137   78-137#  80-53    80-53#   80-62    80-62#   81-12    81-12#
         81-22    81-22#   81-84    81-84#   82-16    82-16#   82-21    82-21#
M$GENB    1-C38#   7-144#  23-17    23-17#   23-23    23-23#   23-25    23-25#   23-52    23-52#   23-56    23-56#   23-60    23-60#
         23-64    23-64#   23-71    23-71#   23-73    23-73#   23-81    23-81#   23-118   23-118#  71-41    71-41#   71-54    71-54#
         71-61    71-61#   71-70    71-70#   71-78    71-78#   71-80    71-80#   71-85    71-85#   71-87    71-87#   71-92    71-92#
         71-98    71-98#   71-103   71-103#
M$GETS    1-D35#   7-144#   9-25     9-25#    10-20    10-20#   10-21    10-21#   15-14    15-14#   15-25    15-25#   15-34    15-34#
         15-40    15-40#   15-49    15-49#   68-1     68-1#    69-76    69-76#   70-12    70-12#   72-124   72-124#  73-18    73-18#
         74-62    74-62#   75-34    75-34#   76-35    76-35#   76-36    76-36#   78-137   78-137#  79-1     79-1#    80-62    80-62#
         81-22    81-22#   81-85    81-85#
M$GETT    1-B77#   7-144#  69-61#   71-16#   71-22#   72-11#   72-109#  74-17#   74-47#   75-19#   76-20#
M$GNGB    1-C02#   7-144#   7-170#   7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
          7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198
          7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198    7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#
          7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   7-198#   8-8      8-8#     9-10     9-10#
         10-10    10-10    10-10#   11-51#   14-17    14-17#   14-27    14-27#   15-9     15-9#    15-16    15-16#   15-27    15-27#
         15-36    15-36#   15-42    15-42#   69-41#   69-47    69-47#   70-8     70-8#    71-8     71-8#    73-10    73-10#   74-8
```

	74 80	75-8	75-80	76-9	76 90	77 500	80 430	80-53	80-530	81-12	81 120	81 84	81 840	
M$GNIN	1 D490	7 1440	7-198	7-198	7-198	7-198	7-190	7 198	7-198	7-198	7-198	7-198	7 198	7-198
	7 198	7-198	7 198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7-198	7 198	7-198	7 198
	7 198	7-198	7-198	7-198	7 198	7 198	7 198	7-198	7-198	7 198	7 198	7 198	7-198	7 198
	7 198	7-198	7-198	7-198	7 198	7 198	7-198	7-1980	7 1980	7-1980	7 1980	7-1980	7 1980	7-1980
	7 1980	7 1980	7-1980	7-1980	7-1980	7-1980	7 1980	7-1980	7-1980	7-1980	7-1980	7 1980	7-1980	7-1980
	7 1980	7-1980	7-1980	7-1980	7-1980	7-1980	7-1980	7-1980	7-1980	7 1980	7-1980	7-1980	7 1980	7-1980
	7 1980	7-1980	7-1980	7-1980	7 1980	7-1980	7-1980	7-1980	8-8	8 8	8 80	8-80	9-10	9-100
	10 10	10-100	14-17	14-17	14-170	14-170	14-27	14-27	14-270	14-270	15-13	15-13	15-13	15-13
	15 13	15-130	15-130	15-130	15-130	15-14	15-140	5-19	15-19	15-19	15-19	15-19	15-19	15-19
	15 19	15-190	15-190	15-190	15-190	15-190	15-190	15-190	15-21	15-21	15-21	15-21	15-21	15-21
	15-21	15-21	15-210	15-210	15-210	15-210	15-210	15-210	15-210	15-24	15-24	15-24	15-24	15-24
	15 240	15-240	15-240	15-240	15-25	15-250	15-30	15-30	15-30	15-30	15-30	15-300	15-300	15-300
	15-300	15-31	15-31	15-31	15-31	15-31	15-31	15 31	15-31	15-31	15-31	15-31	15-31	15 310
	15 310	15-310	15-310	15-310	15-310	15-310	15-310	15 310	15-310	15-310	15-33	15-33	15-33	15-33
	15-33	15-330	15-330	15-330	15-330	15-34	15-340	15-39	15-39	15-39	15-39	15-39	15-390	15-390
	15-390	15-390	15-40	15-400	15-45	15-45	15-45	15-45	15-45	15-45	15-45	15-450	15-450	15-450
	15-450	15-450	15-450	15-48	15-48	15-48	15-48	15-48	15-480	15-480	15-480	15-480	15-49	15-490
	18 16	18-16	18-16	18-16	18-16	18-16	18-160	18-160	18-160	18-160	18-160	18-20	18-20	18-20
	18-20	18-20	18-20	18-20	18-200	18-200	18-200	18-200	18-200	18-200	19-16	19-16	19-16	19-16
	19-16	19-160	19-160	19-160	19-160	19-17	19-17	19-1	19-17	19-17	19-17	19-17	19-17	19-17
	19-17	19-17	19-17	19-170	19-170	19-170	19-170	19-170	19-170	19-170	19-170	19-170	19-170	19-170
	19-19	19-19	19-19	19-19	19-19	19-190	19-190	19-190	19-190	19-20	19-20	19-20	19-20	19-20
	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-200	19-200	19-200	19-200	19-200	19-200	19-200
	19-200	19-200	19-200	19-200	19-22	19-22	19-22	19-22	19-22	19-220	19-220	19-220	19-220	19-23
	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-23	19-230	19-230	19-230
	19-230	19-230	19-230	19-230	19-230	19-230	19-230	19-230	19-28	19-28	19-28	19-28	19-28	19-280
	19-280	19-280	19-280	19-29	19-29	19-29	19-29	19-29	19-29	19-290	19-290	19-290	19-290	19-290
	19-290	19-290	20-38	20-38	20-38	20-38	20-38	20-38	20-380	20-380	20-380	20-380	20-380	23-17
	23-17	23-17	23-17	23-17	23-17	23-17	23-17	23-170	23-170	23-170	23-170	23-23	23-23	23-23
	23-23	23-23	23-23	23-23	23-23	23-230	23-230	23-230	23-230	23-25	23-25	23-25	23-25	23-25
	23-25	23-25	23-25	23-250	23-250	23-250	23-250	23-50	23-50	23-50	23-50	23-50	23-500	23-500
	23-500	23-500	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-52	23-520	23-520	23-520	23-520
	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-56	23-560	23-560	23-560	23-560	23-60	23-60
	23-60	23-60	23-60	23-60	23-60	23-60	23-600	23-600	23-600	23-600	23-64	23-64	23-64	23-64
	23-64	23-64	23-64	23-64	23-640	23-640	23-640	23-640	23-71	23-71	23-71	23-71	23-71	23-71
	23-71	23-71	23-710	23-710	23-710	23-710	23-73	23-73	23-73	23-73	23-73	23-73	23-73	23-73
	23-730	23-730	23-730	23-730	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-81	23-810	23-810
	23-810	23-810	23-118	23-118	23-118	23-118	23-118	23-118	23-1180	23-1180	23-1180	23-1180	24-11	24-11
	24-11	24-11	24-11	24-11	24-11	24-11	24-110	27-38	27-38	27-38	27-38	27-380	27-380	27-380
	27-380	27-380	27-42	27-42	27-42	27-42	27-420	27-420	27-420	27-420	27-420	27-45	27-45	27-45
	27-45	27-450	27-450	27-450	27-450	27-450	27-48	27-48	27-48	27-48	27-480	27-480	27-480	27-480
	27-480	27-51	27-51	27-51	27-51	27-510	27-510	27-510	27-510	27-510	27-54	27-54	27-54	27-54
	27-540	27-540	27-540	27-540	27-540	27-57	27-57	27-57	27 57	27-570	27-570	27-570	27-570	27-570
	27-60	27-60	27-60	27-60	27-600	27-600	27-600	27-600	27 600	27-63	27-63	27-63	27-63	27-630
	27-630	27-630	27-630	27-630	27-66	27-66	27-66	27-66	27-660	27-660	27-660	27-660	27-660	27-69
	27-69	27-69	27-69	27-690	27-690	27-690	27-690	27-690	27-72	27-72	27-72	27-72	27-720	27-720
	27-720	27-720	27-720	27-75	27-75	27-75	27-75	27-750	27-750	27-750	27-750	27-750	27-78	27-78
	27-78	27-78	27-780	27-780	27-780	27-780	27-780	27-81	27-810	27-81	27-81	27-810	27-810	27-810
	27-810	27-810	27-84	27-84	27-84	27-84	27-840	27-840	27-840	27-840	27-840	34-13	34-13	34-13
	34-13	34-13	34-13	34-130	34-130	34-130	34-130	34-130	34-130	34-18	34-18	34-18	34-180	34-180
	34-180	34-180	34-180	34-25	34-25	34-25	34-25	34-25	34-250	34-250	34-250	34-250	34-28	34-28
	34-28	34-28	34-28	34-280	34-280	34-280	34-280	34-34	34-340	34-34	34-34	34-340	34-340	34-340
	34-340	34-340	35-21	35-21	35-21	35-210	35-210	35-210	35-220	35-220	38-45	38-45	38-45	38-45
	38-45	38-45	38-45	38-45	38-450	38-450	38-450	38-450	38-450	38-450	38-450	38-57	38-57	38-57
	38-57	38-57	38-570	38-570	38-570	38-570	38-61	38-61	38-61	38-61	38-61	38-61	38-610	38 610

38 61# 38-61# 38-61# 38-63 38-63 38-63 38-63 38-63 38-63# 38-63# 38-63# 38-63# 39-20 39-20
39 20 39 20 39-20 39-20 39-20# 39-20# 39-20# 39-20# 39-20# 39-22 39-22 39-22 39 22 39-22
39-22# 39-22# 39-22# 39-22# 39-33 39-33 39-33 39-33 39-33 39-33 39-33 39-33 39-33 39-33
39 33# 39-33# 39-33# 39-33# 39-33# 39-33# 39-33# 39-33# 39-33# 39-33# 39-39 39-39 39-39 39-39 39-39
39-39 39-39# 39-39# 39-39# 39-39# 39-39# 39-42 39-42 39-42 39-42 39-42 39-42 39-42# 39-42#
39-42# 39-42# 39-42# 39-46 39-46 39-46 39-46 39-46 39-46# 39-46# 39-46# 39-46# 42-38 42-38
42-38 42-38 42-38 42-38 42-38# 42-38# 42-38# 42-38# 42-38# 47-38 47-38 47-38 47-38 47-38
47-38 47-38 47-38 47-38# 47-38# 47-38# 47-38# 47-38# 47-38# 47-38# 51-43 51-43 51-43 51-43
51-43 51-43 51-43 51-43# 51-43# 51-43# 51-43# 51-43# 51-43# 51-43# 51-48 51-48 51-48 51-48 51-48
51 48 51-48# 51-48# 51-48# 51-48# 51-48# 51-51 51-51 51-51 51-51 51-51 51-51 51-51# 51-51#
51-51# 51-51# 51-51# 51-55 51-55 51-55 51-55 51-55 51-55# 51-55# 51-55# 51-55# 52-21 52-21
52-21 52-21 52-21 52-21 52-21# 52-21# 52-21# 52-21# 52-21# 52-24 52-24 52-24 52-24 52-24
52-24# 52-24# 52-24# 52-24# 52-41 52-41 52-41 [illegible] 52-41 52-41 52-41 52-41 52-41 52-41
52-41# 52-41# 52-41# 52-41# 52-41# 52-41# 52-41# [illegible] 52-41# 52-45 52-45 52-45 52-45 52-45
52-45 52-45# 52-45# 52-45# 52-45# 52-45# 52-48 [illegible] 52-48 52-48 52-48 52-48 52-48# 52-48#
52 48# 52-48# 52-48# 52-53 52-53 52-53 52-53 [illegible] 52-53# 52-53# 52-53# 52-53# 69-61 69-61
69-61# 69-61# 69-7o 69-76# 71-10 71-10# 71-12 [illegible] 71-12# 71-12# 71-14 71-14# 71-16 71-16
71-16# 71-16# 71-18 71-18 71-18# 71-18# 71-20 71-20# 71-22 71-22 71-22# 71-22# 71-24 71-24
71-24# 71-24# 71-26 71-26# 71-28 71-28 71-28# 71-28# 71-30 71-30# 71-37 71-37 71-37 71-37
71-37 71-37# 71-37# 71-37# 71-37# 71-39 71-39 71-39 71-39 71-39 71-39# 71-39# 71-39# 71-39# 71-39#
71-41 71-41 71-41 71-41 71-41 71-41 71-41 71-41 71-41# 71-41# 71-41# 71-41# 71-54 71-54
71-54 71-54 71-54 71-54 71-54# 71-54# 71-54# 71-54# 71-61 71-61 71-61 71-61 71-61 71-61
71-61# 71-61# 71-61# 71-61# 71-70 71-70 71-70 71-70 71-70 71-70 71-70# 71-70# 71-70# 71-70#
71-78 71-78 71-78 71-78 71-78 71-78 71-78 71-78 71-78# 71-78# 71-78# 71-78# 71-80 71-80
71-80 71-80 71-80 71-80 71-80 71-80 71-80# 71-80# 71-80# 71-80# 71-85 71-85 71-85 71-85
71-85 71-85 71-85 71-85# 71-85# 71-85# 71-85# 71-85# 71-87 71-87 71-87 71-87 71-87 71-87
71-87 71-87 71-87# 71-87# 71-87# 71-87# 71-92 71-92 71-92 71-92 71-92 71-92 71-92# 71-92#
71-92# 71-92# 71-98 71-98 71-98 71-98 71-98 71-98 71-98# 71-98# 71-98# 71-98# 71-98# 71-103 71-103
71-103 71-103 71-103 71-103 71-103# 71-103# 71-103# 71-103# 72-10 72-10# 72-11 72-11 72-11# 72-11#
72-13 72-13 72-13# 72-13# 72-109 72-109 72-109# 72-109# 72-124 72-124# 73-18 73-18# 74-10 74-10
74-10# 74-10# 74-14 74-14 74-14 74-14 74-14 74-14# 74-14# 74-14# 74-14# 74-14# 74-17 74-17 74-17#
74-17# 74-21 74-21 74-21 74-21 74-21 74-21 74-21 74-21# 74-21# 74-21# 74-21# 74-21# 74-21# 74-21#
74-24 74-24 74-24 74-24 74-24 74-24 74-24 74-24# 74-24# 74-24# 74-24# 74-24# 74-24# 74-27
74-27 74-27 74-27 74-27 74-27 74-27 74-27# 74-27# 74-27# 74-27# 74-27# 74-27# 74-32 74-32
74-32 74-32 74-32 74-32 74-32 74-32# 74-32# 74-32# 74-32# 74-32# 74-32# 74-34 74-34 74-34
74-34 74-34 74-34 74-34 74-34# 74-34# 74-34# 74-34# 74-34# 74-36 74-36 74-36 74-36 74-36 74-36
74-36 74-36# 74-36# 74-36# 74-36# 74-36# 74-36# 74-47 74-47 74-47# 74-47# 74-62 74-62# 75-19
75-19 75-19# 75-19# 75-34 75-34# 76-20 76-20 76-20# 76-20# 76-35 76-35# 77-106 77-106# 77-119
77-119# 77-120 77-120# 78-14 78-14# 78-137 78-137# 80-53 80-53# 80-55 80-55 80-55 80-55 80-55#
80-57 80-57 80-57 80-57 80-57# 80-59 80-59 80-59 80-59 80-59 80-59# 80-61 80-61 80-61
80-61 80-61 80-61# 80-62 80-62# 81-12 81-12# 81-22 81-22# 81-84 81-84 81-84 81-84# 82-16
82-16 82-16# 82-16#

M$GNLS 1-C13# 7-144# 23-17 23-17# 23-23 23-23# 23-25 23-25# 23-52 23-52# 23-56 23-56# 23-60 23-60#
23-64 23-64# 23-71 23-71# 23-73 23-73# 23-81 23-81# 23-118 23-118# 71-41 71-41# 71-54 71-54#
71-61 71-61# 71-70 71-70# 71-78 71-78# 71-80 71-80# 71-85 71-85# 71-87 71-87# 71-92 71-92#
71-98 71-98# 71-103 71-103#

M$GNSU 1-B98# 7-144#

M$GNTA 1-B90# 7-144# 9-25 9-25# 10-20 10-20# 15-14 15-14# 15-25 15-25# 15-34 15-34# 15-40 15-40#
15-49 15-49# 69-76 69-76# 72-124 72-124# 73-18 73-18# 74-62 74-62# 75-34 75-34# 76-35 76-35#
78-137 78-137# 80-62 80-62# 81-22 81-22# 82-16 82-16# 82-21 82-21#

M$GNTE 1-B94# 7-144# 77-66 77-66#

M$HAPT 1-A39# 7-144# 7-198 7-198#

M$HNAP 1-B24# 7-144# 7-198 7-198#

M$INCR 1-D26# 7-144# 7-170 7-170# 9-10 9-10 9-10# 9-10# 10-10 10-10 10-10# 10-10# 11-51 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# 15-24# 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# 15-40#

```
                15-42    15-42    15-42#   15-42#   15-45#   15-48#   15-49#   18-16#   18-20#   19-16#   19-17#   19-19#   19-20#   19-22#
                19 23#   19-28#   19-29#   20-38#   23-17    23-17#   23-17#   23-23    23-23#   23-23#   23-25    23-25#   23-25#   23-50#
                23-52    23-52#   23-52#   23-56    23-56#   23-56#   23-60    23-60#   23-60#   23-64    23-64#   23-64#   23-71    23-71#
                23-71#   23-73    23-73#   23-73#   23-81    23-81#   23-81#   23-118   23-118#  23-118#  27-38#   27-42#   27-45#   27-48#
                27-51#   27-54#   27-57#   27-60#   27-63#   27-66#   27-69#   27-72#   27-75#   27-78#   27-81#   27-84#   34-13#   34-18#
                34-25#   34-28#   34-34#   35-21#   38-45#   38-57#   38-61#   38-63#   39-20#   39-22#   39-33#   39-39#   39-42#   39-46#
                42-38#   47-38#   51-43#   51-48#   51-51#   51-55#   52-21#   52-24#   52-41#   52-45#   52-48#   52-53#   69-41    69-41#
                69-47    69-47    69-47#   69-47#   69-76#   70-8     70-8     70-8#    70-8#    71-8     71-8     71-8#    71-8#    71-10#
                71-12#   71-16#   71-18#   71-22#   71-24#   71-28#   71-37#   71-39#   71-41    71-41#   71-41#   71-54    71-54#   71-54#
                71-61    71-61#   71-61#   71-70    71-70#   71-70#   71-78    71-78#   71-78#   71-80    71-80#   71-80#   71-85    71-85#
                71-85#   71-87    71-87#   71-87#   71-92    71-92#   71-92#   71-98    71-98#   71-98#   71-103   71-103#  71-103#  72-10#
                72-11#   72-13#   72-109#  72-124#  73-10    73-10    73 10#   73-10#   73-18#   74-8     74-8     74-8#    74-8#    74-10#
                74-14#   74-17#   74-21#   74-24#   74-27#   74-32#   74-34#   74-36#   74-47#   74-62#   75-8     75-8     75-8#    75-8#
                75-34#   76-9     76-9     76-9#    76-9#    76-35#   77-50    77-50#   77-66    77-66    77-66    77-66#   77-66#   77-66#
                77-106#  77-119#  78-14#   78-137#  80-43    80-43#   80-53    80-53    80-53#   80-53#   81-12    81-12    81-12#   81-12#
                82-15    82-15#   82-16    82-16    82-16    82-16#
M$IOSE           1-A00#   7-144#
M$LDRO           1-C42#   7-144#   35-21    35-21#   71-12    71-12#   71-18    71-18#   71-24    71-24#   71-28    71-28#   72-13    72-13#
                74-10    74-10#
M$MASK           1-871#   7-144#
M$MCHI           1-4#     7-144    7-144#   7-144#
M$MCLO           1-824#   7-144    7-144#   7-144#
M$MSK1           1-877#   7-144#
M$POP            1-881#   7-144#   9-25     9-25#    10-20    10-20#   10-21    10-21#   15-14    15-14#   15-25    15-25#   15-34    15-34#
                15-40    15-40#   15-49    15-49#   68-1     68-1#    69-76    69-76#   70-12    70-12#   72-124   72-124#  73-18    73-18#
                74-62    74-62#   75-34    75-34#   76-35    76-35#   76-36    76-36#   78-137   78-137#  79-1     79-1#    80-62    80-62#
                81-22    81-22#   81-85    81-85#
M$PRIN           1-836#   7-144#   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#   19-16    19-16#
                19-17    19-17#   19-19    19-19#   19-20    19-20#   19-22    19-22#   19-23    19-23#   19-28    19-28#   19-29    19-29#
                20-38    20-38#   23-50    23-50#   34-13    34-13#   34-25    34-25#   34-28    34-28#   38-45    38-45#   38-57    38-57#
                38-61    38-61#   38-63    38-63#   39-20    39-20#   39-22    39-22#   39-33    39-33#   39-39    39-39#   39-42    39-42#
                39-46    39-46#   42-38    42-38#   47-38    47-38#   51-43    51-43#   51-48    51-48#   51-51    51-51#   51-55    51-55#
                52-21    52-21#   52-24    52-24#   52-41    52-41#   52-45    52-45#   52-48    52-48#   52-53    52-53#   71-37    71-37#
                71-39    71-39#   74-14    74-14#   74-21    74-21#   74-24    74-24#   74-27    74-27#   74-32    74-32#   74-34    74-34#
                74-36    74-36#
M$PUSH           1-831#   7-144#   7-170    7-170#   9-10     9-10#    10-10    10-10#   11-51    11-51#   15-9     15-9#    15-16    15-16#
                15-27    15-27#   15-36    15-36#   15-42    15-42#   69-41    69-41#   69-47    69-47#   70-8     70-8#    71-8     71-8#
                73-10    73-10#   74-8     74-8#    75-8     75-8#    76-9     76-9#    77-50    77-50#   77-66    77-66#   80-43    80-43#
                80-53    80-53#   81-12    81-12#
M$PUT            1-C72#   7-144#   15-13    15-13    15-13#   15-19    15-19    15-19    15-19    15-19    15-19#   15-21    15-21    15-21
                15-21    15-21    15-21#   15-24    15-24    15-24#   15-30    15-30    15-30#   15-31    15-31    15-31    15-31    15-31
                15-31    15-31    15-31    15-31    15-31#   15-33    15-33    15-33#   15-39    15-39    15-39#   15-45    15-45    15-45
                15-45    15-45#   15-48    15-48    15-48#   18-16    18-16    18-16    18-16#   18-20    18-20    18-20    18-20    18-20#
                19-16    19-16    19-16#   19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17    19-17#   19-19
                19-19    19-19#   19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20    19-20#   19-22    19-22
                19-22#   19-23    19-23    19-23    19-23    19-23    19-23    19-23    19-23    19-23    19-23#   19-28    19-28    19-28#
                19-29    19-29    19-29    19-29    19-29#   20-38    20-38    20-38    20-38#   23-50    23-50    23-50#   34-13    34-13
                34-13    34-13#   34-25    34-25    34-25#   34-28    34-28    34-28#   38-45    38-45    38-45    38-45    38-45    38-45#
                38-57    38-57    38-57#   38-61    38-61    38-61    38-61#   38-63    38-63    38-63#   39-20    39-20    39-20    39-20#
                39-22    39-22    39-22#   39-33    39-33    39-33    39-33    39-33    39-33    39-33    39-33#   39-39    39-39    39-39
                39-39#   39-42    39-42    39-42    39-42#   39-46    39-46    39-46#   42-38    42-38    42-38    42-38#   47-38    47-38
                47-38    47-38    47-38    47-38#   51-43    51-43    51-43    51-43    51-43#   51-48    51-48    51-48    51-48#   51-51
                51-51    51-51    51-51#   51-55    51-55    51-55#   52-21    52-21    52-21    52-21#   52-24    52-24    52-24#   52-41
                52-41    52-41    52-41    52-41    52-41    52-41    52-41#   52-45    52-45    52-45    52-45#   52-48    52-48    52-48
                52 48#   52-53    52-53    52-53#   71-37    71-37    71-37#   71-39    71-39    71-39#   74-14    74-14    74-14#   74-21
```

	74-21	74-21	74-21	74-21#	74-24	74 24	74 24	74-24	74-24#	74-27	74-27	74-27	74-27	74-27#
	74-32	74-32	74-32	74-32	74-32#	74-34	74-34	74-34	74-34#	74-36	74-36	74-36	74 36	74-36#
M$PUT1	1-C81#	7-144#	15-13	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#	15-21	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#	15-31#	15-33	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#	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#	18-20#	19-16	19-16	19-16#	19-16#	19-17	19-17	19-17	19-17	19-17	19-17	19-17	19-17
	19-17	19-17#	19-17#	19-17#	19-17#	19-17#	19-17#	19-17#	19-17#	19-17#	19-19	19-19	19-19#	19-19#
	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20	19-20#	19-20#	19-20#	19-20#	19-20#
	19-20#	19-20#	19-20#	19-20#	19-22	19-22	19-22#	19-22#	19-23	19-23	19-23	19-23	19-23	19-23
	19-23	19-23	19-23	19-23#	19-23#	19-23#	19-23#	19-23#	19-23#	19-23#	19-23#	19-23#	19-28	19-28
	19-28#	19-28#	19-29	19-29	19-29	19-29	19-29#	19-29#	19-29#	19-29#	20-38	20-38	20-38	20-38#
	20-38#	20-38#	23-50	23-50	23-50#	23-50#	34-13	34-13	34-13	34-13#	34-13#	34-13#	34-25	34-25
	34-25#	34-25#	34-28	34-28	34-28#	34-28#	38-45	38-45	38-45	38-45	38-45	38-45#	38-45#	38-45#
	38-45#	38-45#	38-57	38-57	38-57#	38-57#	38-61	38-61	38-61	38-61#	38-61#	38-61#	38-63	38-63
	38-63#	38-63#	39-20	39-20	39-20	39-20#	39-20#	39-20#	39-22	39-22	39-22#	39-22#	39-33	39-33
	39-33	39-33	39-33	39-33	39-33	39-33#	39-3?#	39-33#	39-33#	39-33#	39-33#	39-33#	39-39	39-39
	39-39	39-39#	39-39#	39-39#	39-42	39-42	39-42	39-42#	39-42#	39-42#	39-46	39-46	39-46#	39-46#
	42-38	42-38	42-38	42-38#	42-38#	42-38#	47-38	47-38	47-38	47-38	47-38	47-38#	47-38#	47-38#
	47-38#	47-38#	51-43	51-43	51-43	51-43	51-43#	51-43#	51-43#	51-43#	51-48	51-48	51-48	51-48#
	51-48#	51-48#	51-51	51-51	51-51	51-51#	51-51#	51-51#	51-55	51-55	51-55#	51-55#	52-21	52-21
	52-21	52-21#	52-21#	52-21#	52-24	52-24	52-24#	52-24#	52-41	52-41	52-41	52-41	52-41	52-41
	52-41	52-41#	52-41#	52-41#	52-41#	52-41#	52-41#	52-41#	52-45	52-45	52-45	52-45#	52-45#	52-45#
	52-48	52-48	52-48	52-48#	52-48#	52-48#	52-53	52-53	52-53#	52-53#	71-37	71-37	71-37#	71-37#
	71-39	71-39	71-39#	71-39#	74-14	74-14	74-14#	74-14#	74-21	74-21	74-21	74-21	74-21#	74-21#
	74-21#	74-21#	74-24	74-24	74-24	74-24	74-24#	74-24#	74-24#	74-24#	74-27	74-27	74-27	74-27
	74-27#	74-27#	74-27#	74-27#	74-32	74-32	74-32	74-32	74-32#	74-32#	74-32#	74-32#	74-34	74-34
	74-34	74-34#	74-34#	74-34#	74-36	74-36	74-36	74-36	74-36#	74-36#	74-36#	74-36#		
M$RADI	1-D77#	7-144#	23-17	23-17#	23-23	23-23#	23-25	23-25#	23-52	23-52#	23-56	23-56#	23-60	23-60#
	23-64	23-64#	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	71-41	71-41#	71-54	71-54#
	71-61	71-61#	71-70	71-70#	71-78	71-78#	71-80	71-80#	71-85	71-85#	71-87	71-87#	71-92	71-92#
	71-98	71-98#	71-103	71-103#	80-55	80-55#	80-57	80-57#	80-59	80-59#	80-61	80-61#		
M$RBRO	1-C52#	7-144#												
M$RNRO	1-C62#	7-144#	35-21	35-21#										
M$SETS	1-D32#	7-144#	7-170	7-170#	9-10	9-10#	10-10	10-10#	11-51	11-51#	15-9	15-9#	15-16	15-16#
	15-27	15-27#	15-36	15-36#	15-42	15-42#	69-41	69-41#	69-47	69-47#	70-8	70-8#	71-8	71-8#
	73-10	73-10#	74-8	74-8#	75-8	75-8#	76-9	76-9#	77-50	77-50#	77-66	77-66#	80-43	80-43#
	80-53	80-53#	81-12	81-12#										
M$STAR	1-A33#	7-144#												
M$SVC	1-C33#	7-144#	15-13	15-13#	15-14	15-14#	15-19	15-19#	15-21	15-21#	15-24	15-24#	15-25	15-25#
	15-30	15-30#	15-31	15-31#	15-33	15-33#	15-34	15-34#	15-39	15-39#	15-40	15-40#	15-45	15-45#
	15-48	15-48#	15-49	15-49#	18-16	18-16#	18-20	18-20#	19-16	19-16#	19-17	19-17#	19-19	19-19#
	19-20	19-20#	19-22	19-22#	19-23	19-23#	19-28	19-28#	19-29	19-29#	20-38	20-38#	23-17	23-17#
	23-23	23-23#	23-25	23-25#	23-50	23-50#	23-52	23-52#	23-56	23-56#	23-60	23-60#	23-64	23-64#
	23-71	23-71#	23-73	23-73#	23-81	23-81#	23-118	23-118#	27-38	27-42	27-45	27-48	27-51	27-54
	27-57	27-60	27-63	27-66	27-69	27-72	27-75	27-78	27-81	27-84	34-13	34-13#	34-18	34-25
	34-25#	34-28	34-28#	34-34	35-21	35-21#	38-45	38-45#	38-57	38-57#	38-61	38-61#	38-63	38-63#
	39-20	39-20#	39-22	39-22#	39-33	39-33#	39-39	39-39#	39-42	39 ?2#	39-46	39-46#	42-38	42-38#
	47-38	47-38#	51-43	51-43#	51-48	51-48#	51-51	51-51#	51-55	51-55#	52-21	52-21#	52-24	52-24#
	52-41	52-41#	52-45	52-45#	52-48	52-48#	52-53	52-53#	69-61#	69-76	69-76#	71-10	71-10#	71-12
	71-12#	71-16	71-16#	71-18	71-18#	71-22	71-22#	71-24	71-24#	71-28	71-28#	71-37	71-37#	71-39
	71-39#	71-41	71-41#	71-54	71-54#	71-61	71-61#	71-70	71-70#	71-78	71-78#	71-80	71-80#	71-85
	71-85#	71-87	71-87#	71-92	71-92#	71-98	71-98#	71-103	71-103#	72-10	72-10#	72-11	72-11#	72-13
	72-13#	72-109	72-109#	72-124	72-124#	73-18	73-18#	74-10	74-10#	74-14	74-14#	74-17	74-17#	74-21

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