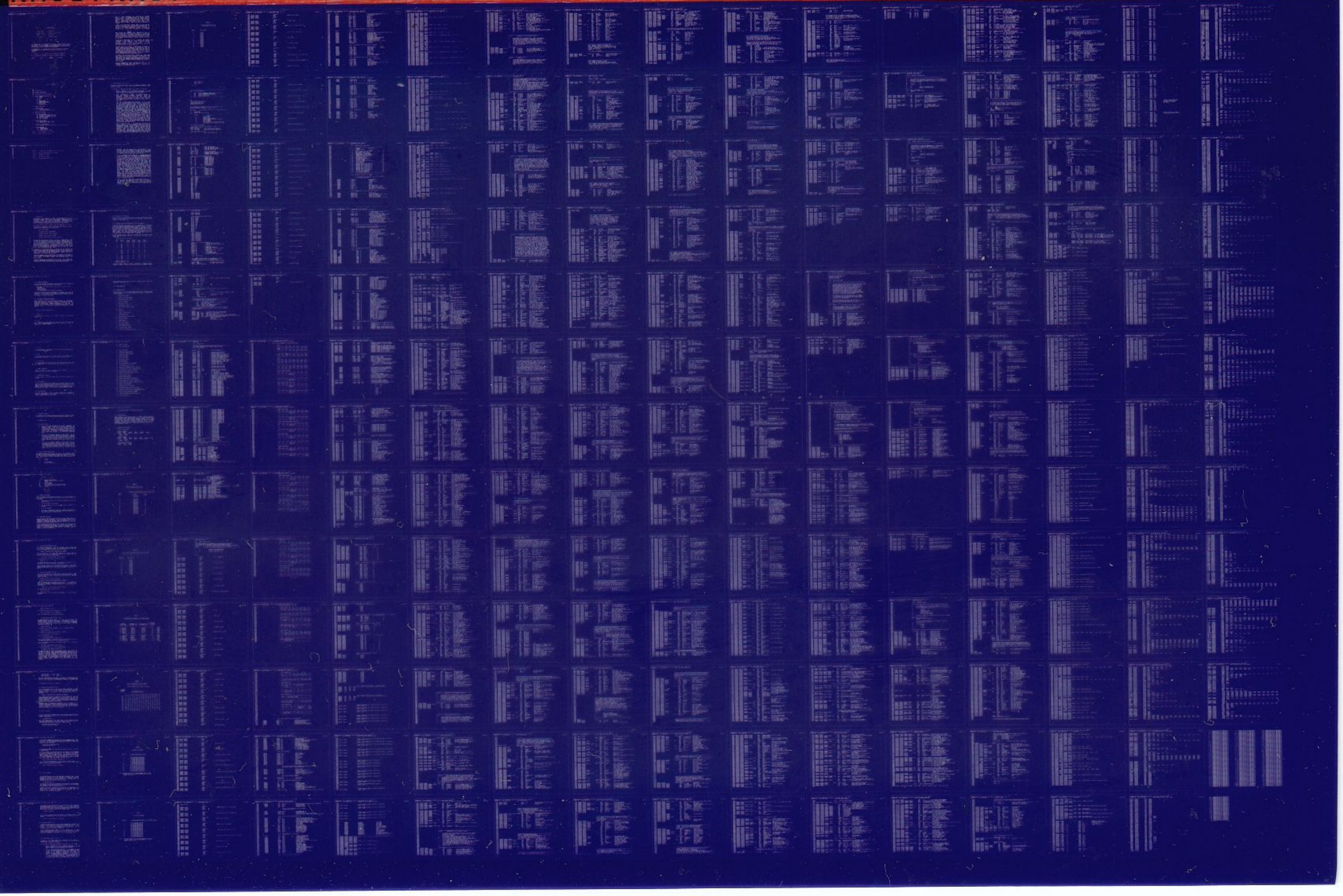
RK611
RK06,RK07

RK6 DR CPT PROG
CZR6QC0

AH-B162C-MC
FICHE 1 OF 2

NOV 1980
COPYRIGHT@ 77-80
MADE IN USA



RK611 RK6 DR CPT PROG AH-B162C-MC NOV 1980
RK06,RK07 CZR6QC0 FICHE 2 OF 2 COPYRIGHT@ 77-80
MADE IN USA

```
.REM @
```

IDENTIFICATION

PRODUCT CODE:	AC-B161C-MC
PRODUCT NAME:	CZR6QC0 RK6 DR CPT PROG
DATE:	JUNE 1980
MAINTAINER:	DIAGNOSTIC GROUP

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

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

COPYRIGHT (C) 1977,1980 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

TABLE OF CONTENTS

```
103
104
105
106                          TABLE C -  CYLINDER BLOCK ASSIGNMENT FOR A GIVEN SURFACE
107
108                          TABLE D -  BASIC CYLINDER BLOCK LAYOUT EXAMPLE
109
110                          TABLE E -  OVERWRITE CYLINDERS
111
112                          TABLE F -  SELF-TEST CYLINDERS
113
114                          TABLE G -  PSEUDO-RANDOM DATA PATTERN
```

1.0 INTRODUCTION

THE PURPOSE OF THIS PROGRAM IS TO VERIFY THE COMPATIBILITY OF UP TO 16 RK06 OR RK07 DRIVES WHICH MAY RESIDE ON ONE OR MORE RK611/RK06-07 SUBSYSTEMS (INCLUDING SYSTEMS WITH MULTIPLE SUBSYSTEMS). COMPATIBILITY IS DEFINED HERE AS THE ABILITY OF A DRIVE TO WRITE DATA WHICH CAN BE READ SUCCESSFULLY BY ALL OTHER DRIVES, AND ADDITIONALLY THE ABILITY OF A DRIVE TO COMPLETELY OVER-WRITE DATA WRITTEN BY ALL OTHER DRIVES.

NOTE: RK06 AND RK07 DRIVES CANNOT BE MIXED FOR COMPATIBILITY TESTING.

THE PROGRAM IS DESIGNED TO DETECT THE FOLLOWING CONDITIONS WHICH MOST COMMONLY CAUSE INCOMPATIBILITY BETWEEN DRIVES:

1. HEAD MIS-ALIGNMENT
2. POSITIONER LATERAL MISALIGNMENT
3. SPINDLE-CARTRIDGE INTERFACE RUNOUT
4. IMPROPER LEVELS OF WRITE CURRENT
5. INCORRECT ADDRESSING OF READ/WRITE HEADS

THE TESTING IS DONE IN TWO PASSES. IN PASS 1, COMPATIBILITY DATA PATTERNS ARE WRITTEN BY ALL THE DRIVES UPON THE SAME DISK CARTRIDGE, AND THE BASIC READ/WRITE CAPABILITY OF EACH DRIVE IS DEMONSTRATED. IN PASS 2, THE COMPATIBILITY DATA FROM ALL DRIVES IS READ BY EACH DRIVE, WITH INCREASING HEAD OFFSET, AND THIS IS COMPARED WITH EACH DRIVE'S ABILITY TO READ ITS OWN DATA. IN ADDITION, EACH DRIVE'S CAPABILITY TO OVERWRITE DATA WRITTEN BY ALL OTHER DRIVES IS TESTED ON THE SECOND PASS. (FOR THE REMAINDER OF THIS SPECIFICATION, THE ABOVE DEFINITIONS OF THE FIRST AND SECOND PASS SHALL APPLY).

IN BOTH PASSES, THE PROGRAM DIRECTS THE OPERATOR IN THE LOADING AND UNLOADING OF DRIVES AND THE MOVEMENT OF THE CARTRIDGE FROM DRIVE TO DRIVE, THROUGH MESSAGES AT THE CONSOLE TERMINAL. AT THE COMPLETION OF TESTING ON EACH DRIVE DURING THE SECOND PASS A SUMMARY IS PRINTED OF COMPATIBILITY TEST RESULTS FOR THAT DRIVE.

WITHIN THE VARIOUS TESTS OF BOTH PASSES, THE CAPABILITY IS PROVIDED TO LOOP ON CURRENT OPERATIONS, AND SWITCH REGISTER OPTIONS ARE PROVIDED, FOR A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES (SEE SECTION 6.2).

UNEXPECTED ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION, CURRENT AND PREVIOUS OPERATIONS GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE RK06-07 DRIVE COMPATIBILITY PROGRAM, FOR EACH SUBSYSTEM WHICH MAY BE PRESENT:

- PDP-11/04, (05,10 MFG. ONLY), 20,30,34,35,40,45,50,70, OR PDQ
- 16K MEMORY
- CONSOLE TERMINAL
- RK611 CONTROLLER
- 1 TO 8 RK06 OR RK07 DISK DRIVES PER CONTROLLER

IN ADDITION, A SINGLE RK06 DISK CARTRIDGE IS REQUIRED WHICH MAY BE FORMATTED IN EITHER 20 OR 22 SECTOR FORMAT, ON A RELIABLE WELL-ALIGNED RK06 DRIVE. THIS CARTRIDGE WILL BE MOVED FROM DRIVE TO DRIVE, (ON UP TO 16 DRIVES) ON EACH OF TWO PASSES.

3.0 PRELIMINARY PROGRAM REQUIREMENTS

BEFORE RUNNING THE RK06 DRIVE COMPATIBILITY PROGRAM, THE SUBSYSTEM(S) UNDER TEST SHOULD BE CAPABLE OF PASSING THE CONTROLLER DIAGNOSTICS CZR6A-CZR6E AND CZR6K, THE DRIVE DIAGNOSTICS CZR6H-CZR6J, AND THE SUBSYSTEM VERIFICATION PROGRAMS CZR6M-CZR6N. IN ADDITION, THE CARTRIDGE MUST BE FORMATTED IN 20 OR 22 SECTOR FORMAT USING THE PACK FORMATTER CZR6L, OR EQUIVALENT (PERFORMED ON KNOWN GOOD, ALIGNED DRIVE).

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 SYSMAC

THIS PROGRAM USES PORTIONS OF THE SYSMAC DIAGNOSTIC SYSTEM MACRO PACKAGE.

4.2 XXDP

THIS PROGRAM MAY BE LOADED UNDER XXDP, AND MAY BE RUN IN DUMP MODE ONLY. DUE TO MANUAL INTERVENTION AND LACK OF END-OF-PASS HOOKS, THE PROGRAM IS NOT XXDP CHAINABLE.

4.3 ACT

THIS PROGRAM MAY BE LOADED UNDER ACT AND MAY BE RUN IN DUMP MODE ONLY.
IT IS NOT CHAINABLE UNDER ACT.

4.4 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM, BUT MAY BE RUN IN
PROGRAM (DUMP) MODE ONLY. IT CANNOT BE RUN IN APT SCRIPT MODE.

4.5 DUAL-ACCESS

THIS PROGRAM DOES NOT UTILIZE THE DUAL-ACCESS OPTION IN ANY WAY, AND
ALL DRIVES UNDER TEST SHOULD BE DE-SELECTED THROUGH THE PORT WHICH IS
NOT IN USE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS
INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE DISABLED BY THE
PROGRAM.

4.8 BAD SECTORS

THE LIST OF BAD SECTORS ON THE CARTRIDGE IS OBTAINED FROM THE FIRST
DRIVE TO BE TESTED ON THE CURRENT SUBSYSTEM. ACCORDING TO A SWITCH
REGISTER OPTION (SEE SECTION 6.2) THIS LIST MAY BE TYPED AT THE
CONSOLE AT THE START OF THE FIRST PASS. ALL DATA ERRORS OCCURING IN
THE PROGRAM ARE IGNORED IF THEY OCCUR IN SECTORS DESIGNATED AS BAD.

4.9 EXECUTION TIME

THE TOTAL TIME REQUIRED TO RUN THE DRIVE COMPATIBILITY PROGRAM IS
DIRECTLY PROPORTIONAL TO THE NUMBER OF DRIVES TO BE TESTED AND
REQUIRES ABOUT 8-10 MINUTES PER DRIVE.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM THE ACT OR APT SYSTEMS OR FROM ANY MEDIA SUPPORTED BY XXDP.

6.0 PROGRAM OPTIONS

6.1 STARTING ADDRESSES

200 - THIS IS THE STARTING ADDRESS FOR DEFAULT PARAMETERS AND RUNNING OF PASS 1 AND PASS 2 ON A SINGLE SUBSYSTEM. THE PROGRAM WILL USE DEFAULT RK611 BASE ADDRESS, INTERRUPT VECTOR AND PRIORITY, AND WILL AUTOMATICALLY DETERMINE WHICH DRIVES TO TEST. DRIVES MUST BE ON-LINE, POWERED UP, AND EITHER LOADED OR UNLOADED. THE PROGRAM WILL ASSUME ALL DRIVES TO BE TESTED RESIDE ON ONE RK611/RK06 SUBSYSTEM ONLY.

204 - THIS IS THE STARTING ADDRESS TO RUN PASS 1 ON ALL RK611/RK06 SUBSYSTEMS WHICH RESIDE ON THIS PDP-11 SYSTEM. THE PROGRAM WILL ASK FOR THE RK611 BASE ADDRESS, INTERRUPT VECTOR, AND PRIORITY FOR EACH SUBSYSTEM ON THIS SYSTEM, AND IT ASKS FOR THE LETTER NAMES (A THRU P) ASSIGNED TO ALL OTHER SUBSYSTEMS, AND THE DRIVE(S) WHICH WILL BE TESTED ON EACH.

220 - THIS IS THE STARTING ADDRESS TO RUN PASS 2 ON ALL RK611/RK06 SUBSYSTEMS WHICH RESIDE ON THIS PDP-11 SYSTEM. THE PROGRAM WILL ASK FOR THE RK611 BASE ADDRESS,INTERRUPT VECTOR, AND PRIORITY FOR EACH SUBSYSTEM ON THIS SYSTEM, AND IT ASKS FOR THE LETTER NAMES (A THRU P) ASSIGNED TO ALL OTHER SUBSYSTEMS, AND THE DRIVE(S) WHICH WILL BE TESTED ON EACH.

6.2 SWITCH REGISTER OPTIONS USED

THIS PROGRAM IS DESIGNED TO ALLOW THE USE OF THE HARDWARE SWITCH REGISTER IF PRESENT, OR THE SYSMAC-SUPPORTED SOFTWARE SWITCH REGISTER (IF HARDWARE SWR IS NOT PRESENT, OR IS SET TO ALL ONES). IN EITHER CASE, THE FOLLOWING OPTIONS ARE IMPLEMENTED WHEN THE APPROPRIATE BITS ARE SET TO 1:

BIT	OPTION
15	HALT ON ERROR
14	LOOP ON CURRENT TEST

13	INHIBIT ERROR REPORTS
12	REPORT DESCRIPTION ONLY, ON ERRORS
11	UNUSED
10	BELL ON ERROR
09	LOOP ON ERROR
08	APPLY RANDOM STALL BETWEEN OPERATIONS
07	TYPE BAD SECTOR FILES (BSF'S) AT START
06-00	UNUSED

7.0 RUNNING THE PROGRAM

ONCE THE PROGRAM HAS BEEN LOADED INTO CORE (IN A GIVEN SYSTEM, IF THERE ARE MULTIPLE SYSTEMS) THE FOLLOWING STEPS MUST BE TAKEN TO RUN THE PROGRAM:

1. INSURE THAT ALL DRIVES TO BE TESTED ARE POWERED UP AND SINGLE PORT SELECTED.

2. LOAD THE DESIRED START ADDRESS.

3. SET ANY DESIRED BITS IN THE HARDWARE SWITCH REGISTER (IF PRESENT).

4. START THE PROGRAM.

5. FOLLOW ALL INSTRUCTIONS TYPED BY THE PROGRAM PERTAINING TO THE MANUAL INTERVENTION REQUIRED, AND THE ALTERNATE USE OF MULTIPLE SYSTEMS (IF THERE ARE ANY).

8.0 OPERATIONAL DIALOGUE

THIS SECTION DESCRIBES THE CONSOLE TERMINAL DIALOGUE THROUGH WHICH THE PROGRAM DIRECTS THE OPERATOR, IN THE SELECTION OF OPTIONS AND THE LOADING AND UN_OADING OF DRIVES, AND THE MOVEMENT OF THE TEST CARTRIDGE. THE EXACT DIALOGUE WHICH IS USED DEPENDS UPON THE STARTING ADDRESS WHICH WAS CHOSEN (SEE SECTION 6.1).

IN THE FOLLOWING DISCUSSION AND IN THE PRINTOUT OF TEST RESULTS, DRIVES TO BE TESTED WILL BE REFERRED TO BY A LETTER AND A NUMBER. THE LETTER IS THE SUBSYSTEM LETTER NAME (OPERATOR ASSIGNED), AND THE NUMBER IS THE DRIVE NUMBER ON THAT SUBSYSTEM. FOR EXAMPLE, DRIVE C6 REFERS TO DRIVE 6 ON SUBSYSTEM C.

8.1 DIALOGUE FOR ADDRESS 200 START

THIS STARTING ADDRESS MAY BE USED FOR AUTOMATIC DRIVE SIZING AND DEFAULTING OF PARAMETERS, WHEN THE DRIVES RESIDE ON ONE SUBSYSTEM ONLY. THE PROGRAM FIRST IDENTIFIES ITSELF AS FOLLOWS:

```
     CZR6QB0 - RK06-RK07 DRIVE COMPATIBILITY PROGRAM
```

THEN, THE DRIVES ON THE SYSTEM ARE AUTOMATICALLY SIZED, AND ALL EXISTING DRIVES WILL BE MARKED FOR TESTING. THE PROGRAM TYPES THE DRIVE LIST, AS IN THE FOLLOWING EXAMPLE:

```
        DRIVE TYPE  6<CR> FOR RK06, 7<CR> FOR RK07:
```

THE OPERATOR MUST TYPE EITHER A 6<CR> OR A 7<CR>

```
     DRIVES = 2,5,7
```

THE PROGRAM NOW PROCEEDS WITH PASS 1, AND DIRECTS THE OPERATOR IN THE MOUNTING OF THE PACK, AS DESCRIBED IN SECTION 8.4.

PLEASE NOTE THAT THERE IS ONLY ONE SUBSYSTEM ON AN ADR. 200 START, AND IT IS NAMED SUBSYSTEM A. THE DRIVES IN THE ABOVE EXAMPLE WOULD BE REFERRED TO AS A2,A5,A7 IN THE TEST RESULTS PRINTOUT AT THE END OF PASS 2.

8.2 DIALOGUE FOR ADDRESS 204 START

THIS STARTING ADDRESS MUST BE USED ON EACH SYSTEM, WHEN THERE IS MORE THAN ONE SUBSYSTEM, BUT IT MAY ALSO BE USED WHEN THERE IS JUST ONE SUBSYSTEM (TOTAL), TO SPECIFY DRIVES TO TEST AND NON-DEFAULT PARAMETER VALUES, FOR PASS 1.

THE PROGRAM IDENTIFIES ITSELF AS FOLLOWS:

```
     CZR6QB0 - RK06-RK07 DRIVE COMPATIBILITY PROGRAM
```

THEN, THE PROGRAM ASKS THE OPERATOR FOR THE DRIVES TO BE TESTED ON EACH OF THE POSSIBLE SUBSYSTEMS (STARTING WITH A - THE NAMES RANGE FROM SUBSYS A TO SUBSYS P. THERE COULD BE UP TO 16 SUBSYSTEMS, WITH A DRIVE ON EACH) :

```
        DRIVE TYPE  6<CR> FOR RK06, 7<CR> FOR RK07:
```

THE OPERATOR MUST TYPE IN 6<CR> OR 7<CR>

```
     SUBSYS A DRIVE(S) =
```

THE OPERATOR THEN TYPES THE DESIRED DRIVE NUMBERS, AS IN THE FOLLOWING EXAMPLE:

```
                                        SUBSYS A DRIVE(S) = 2,5,7

                                    THE PROGRAM THEN VERIFIES THE DRIVE NOS.  BY TYPING :

                                        WILL TEST DRIVE(S) 2,5,7 ON SUBSYS A.

                                    NEXT, THE PROGRAM ASKS THE FOLLOWING QUESTION:



                                        IS THERE ANOTHER SUBSYS (Y OR N <CR>)?

                                    THE OPERATOR TYPES Y<CR> OR  N<CR>.   (IF  JUST  <CR>  IS  TYPED,  THE
                                    PROGRAM  ASSUMES  THAT N<CR> WAS TYPED).  IF THE OPERATOR TYPED N, THE
                                    PROGRAM PROCEEDS WITH PASS 1, AND DIRECTS THE OPERATOR IN THE MOUNTING
                                    OF THE PACK, AS DESCRIBED IN SECTION 8.4.  IF Y WAS TYPED, THE PROGRAM
                                    ASKS FOR THE NUMBERS OF THE DRIVES TO BE TESTED ON THE NEXT  SUBSYSTEM
                                    (SUBSYS B) AS FOLLOWS:

                                        SUBSYS B DRIVE(S) =

                                    THE OPERATOR TYPES THE DRIVE NUMBERS, AS IN THE FOLLOWING EXAMPLE:

                                        SUBSYS B DRIVE(S) = 2,3

                                    THE PROGRAM THEN VERIFIES THE DRIVE NOS.  BY TYPING :

                                        WILL TEST DRIVE(S) 2,3 ON SUBSYS B.

                                    NEXT, THE PROGRAM WILL ASK :

                                        IS THERE ANOTHER SUBSYS (Y OR N <CR>)?

                                    AND IN THE SAME MANNER, THE OPERATOR SPECIFIES THE DRIVES ON  EACH  OF
                                    THE REMAINING SUBSYSTEMS, UNTIL ALL HAVE BEEN SPECIFIED.

                                    NEXT, THE PROGRAM ASKS FOR THE LETTER NAME ASSIGNED TO  THE  SUBSYSTEM
                                    TO BE TESTED NEXT :

                                        TYPE NAME OF NEXT SUBSYS TO TEST (A-P) :

                                    THE OPERATOR RESPONDS, AS IN THE FOLLOWING EXAMPLE :

                                        TYPE NAME OF NEXT SUBSYS TO TEST (A-P) :  A

                                    ALL SUBSYSTEMS MUST BE TESTED IN THE ORDER IN WHICH  THE  LETTERS  ARE
                                    ASSIGNED  (A  THRU P).  NEXT, THE PROGRAM ALLOWS THE OPERATOR TO ALTER
                                    THE RK611 BUS ADDRESS, VECTOR ADDRESS, AND INTERRUPT PRIORITY FOR THIS
                                    SUBSYSTEM.   FOR  EACH  PARAMETER  THE CURRENT VALUE IS TYPED, AND THE
                                    OPERATOR IS GIVEN THE OPPORTUNITY TO TYPE IN A NEW VALUE,  PLUS  <CR>.
                                    IF JUST <CR> IS TYPED, THE PARAMETER IS NOT CHANGED.  WHEN THE PROGRAM
                                    IS FIRST LOADED, THE FOLLOWING DEFAULT VALUES ARE ASSIGNED:  RK611 BUS
                                    ADDRESS = 177440,   VECTOR   ADDRESS = 210,   AND  PRIORITY = 5.   THE
                                    FOLLOWING EXAMPLE SHOWS A  PRINTOUT  IN  WHICH  ONLY  THE  VECTOR  WAS
                                    CHANGED:
```

```
                                        RK611 BUS ADR  =  177440    NEW =
                                        RK611 VEC ADR  =     210    NEW = 240
                                        RK611 PRIORITY =       5    NEW =

                                    THEN THE PROGRAM PROCEEDS WITH PASS 1, AND DIRECTS THE OPERATOR IN THE
                                    MOUNTING  OF THE PACK, AS DESCRIBED IN SECTION 8.4.  AT THE COMPLETION
                                    OF PASS 1 ON THE SUBSYSTEM, THE PROGRAM WILL INFORM THE  OPERATOR  HOW



                                    TO PERFORM PASS 1 ON THE NEXT SUBSYSTEM.



                                    8.3  DIALOGUE FOR ADDRESS 220 START

                                    THIS STARTING ADDRESS MUST BE USED ON EACH SYSTEM, WHEN THERE IS  MORE
                                    THAN  1  SUBSYSTEM,  BUT  IT  MAY  ALSO BE USED WHEN THERE IS JUST ONE
                                    SUBSYSTEM  (TOTAL)  ,  TO  SPECIFY  DRIVES  TO  TEST  AND  NON-DEFAULT
                                    PARAMETER  VALUES,  FOR  PASS  2.   THE  PROGRAM IDENTIFIES ITSELF, AS
                                    FOLLOWS :

                                         CZR6QB0 - RK06-RK07 DRIVE COMPATIBILITY PROGRAM

                                    THE DIALOGUE FOR 220 START IS IDENTICAL TO THE DIALOGUE  FOR  THE  204
                                    START  DESCRIBED  ABOVE  (SECTION 8.2), FOR THE SELECTION OF SUBSYSTEM
                                    PARAMETERS AND THE SPECIFICATION OF ALL DRIVES TO  BE  TESTED  ON  THE
                                    VARIOUS  SUBSYSTEMS.   HOWEVER,  AFTER THIS DIALOGUE IS COMPLETED, THE
                                    PROGRAM PROCEEDS WITH PASS 2, AND DIRECTS THE OPERATOR IN THE MOVEMENT
                                    OF THE PACK, AS DESCRIBED IN SECTION 8.5.

                                    NOTE THAT SINCE THE APPROPRIATE  PROCESSOR  MUST  BE  STARTED  AT  THE
                                    STARTING  ADDRESS  FOR  EACH SUBSYSTEM TO BE TESTED, THE COMPATIBILITY
                                    TEST MAY BE PERFORMED IN STEPS, AT VARIOUS TIMES AND  BETWEEN  VARIOUS
                                    DISTANT  LOCATIONS,  BY  MOVING  THE TEST PACK AND SAVING THE PRINTOUT
                                    FROM EACH PASS ON EACH PDP-11 SYSTEM INVOLVED.



                                    8.4  PASS 1 DIALOGUE



                                    AFTER THE SELECTION OF PARAMETERS AND DRIVES HAS BEEN COMPLETED ON THE
                                    CURRENT  SUBSYSTEM (SECTIONS 8.1-8.2), THE PROGRAM INDICATES THE START
                                    OF PASS 1 AS FOLLOWS:

                                         ** STARTING PASS 1 **

                                    NEXT, THE PROGRAM SELECTS  THE  FIRST  DRIVE  TO  BE  TESTED  ON  THIS
                                    SUBSYSTEM,  AND INSTRUCTS THE OPERATOR TO MOUNT THE TEST CARTRIDGE AND
                                    LOAD THE HEADS ON THAT DRIVE, AS IN THE FOLLOWING EXAMPLE:

                                         MOUNT PACK ON DRIVE A2 AND LOAD.
                                         TYPE R<CR> WHEN DRIVE READY:
```

THE OPERATOR PERFORMS THIS TASK AND TYPES R<CR> WHEN THE DRIVE READY LIGHT IS ON. THE PROGRAM PERFORMS PASS 1 FUNCTIONS ON THIS DRIVE (SEE SECTION 9.1) AND THEN INSTRUCTS THE OPERATOR TO UNLOAD THE DRIVE AND REMOVE THE PACK AS FOLLOWS:

```
UNLOAD DRIVE A2 AND REMOVE PACK.
TYPE R<CR> WHEN DONE:
```

THE OPERATOR PERFORMS THESE FUNCTIONS AND TYPES R<CR> AFTER THE PACK HAS BEEN REMOVED.

IN THE SAME MANNER, THE PROGRAM INSTRUCTS THE OPERATOR IN THE MOVEMENT OF THE PACK THROUGHOUT THE REST OF THE DRIVES ON THE CURRENT SUBSYSTEM. WHEN THIS HAS BEEN COMPLETED, THE PROGRAM DOES ONE OF THREE THINGS: (1) IF THERE IS ONLY ONE SUBSYSTEM (FROM ADR 200 START) THE PROGRAM BEGINS PASS 2 (SEE SECTION 8.5). (2) IF THERE IS ANOTHER SUBSYSTEM, THE PROGRAM DIRECTS THE OPERATOR TO PERFORM PASS 1 ON THE NEXT SUBSYS AS FOLLOWS (AND THEN HALTS) :

```
START AT ADR 204 FOR PASS 1 ON SUBSYS B
```

(3) IF THERE ARE NO MORE DRIVES TO TEST IN PASS 1 ON ANY SUBSYS, THE PROGRAM DIRECTS THE OPERATOR TO BEGIN PASS 2 ON THE FIRST SUBSYS (SEE SECT. 8.5) AS FOLLOWS (AND THEN HALTS) :

```
START AT ADR 220 FOR PASS 2 ON SUBSYS A
```

8.5 PASS 2 DIALOGUE

THE OPERATOR RETURNS TO THE FIRST SUBSYSTEM TO PERFORM PASS 2 EITHER THROUGH THE DIALOGUE OF THE ADR 200 START, OR AFTER THE SELECTION OF PARAMETERS AND DRIVES HAS BEEN COMPLETED IN ACCORDANCE WITH THE DIALOGUE OF THE ADR 220 START (SEE SECTION 8.3). IN EITHER CASE, THE PROGRAM INDICATES THE START OF PASS 2 BY TYPING :

```
** STARTING PASS 2 **
```

THE PROGRAM THEN DIRECTS THE OPERATOR IN THE UNLOADING, PACK MOVEMENT, AND LOADING OF ALL DRIVES ON THE FIRST SUBSYSTEM, IN THE SAME MANNER AS DESCRIBED FOR PASS 1 (SEE SECTION 8.4).

HOWEVER, AFTER PASS 2 TESTING (SEE SECTION 9.2) IS COMPLETED ON A GIVEN DRIVE, THE ENTIRE TEST RESULTS FOR THAT DRIVE ARE TYPED. THE DETAILS OF THIS PRINTOUT ARE DESCRIBED IN SECTION 10, AFTER THE DETAILS OF THE TESTING ARE DESCRIBED.

WHEN PASS 2 HAS BEEN COMPLETED FOR ALL DRIVES ON THE FIRST SUBSYSTEM, THE PROGRAM DOES ONE OF TWO THINGS: (1) IF THERE IS ONLY ONE SUBSYSTEM (FROM ADR 200 START) OR IF ALL DRIVES ON ALL SUBSYSTEMS HAVE BEEN TESTED IN PASS 2 (FROM ADR 220 START), THE ENTIRE TESTING AND REPORTING HAVE BEEN COMPLETED, AND THE PROGRAM TYPES:

** END OF TESTING **

(2) IF THERE IS ANOTHER SUBSYSTEM, HOWEVER, THE PROGRAM DIRECTS THE OPERATOR TO PERFORM PASS 2 ON THE NEXT SUBSYSTEM AS FOLLOWS (AND THEN HALTS):

START AT ADR 220 FOR PASS 2 ON SUBSYS B

9.0 DESCRIPTION OF TESTS

THE MAIN FUNCTIONAL BLOCKS OF CODE IN THE PROGRAM ARE ASSIGNED TEST NUMBERS, FOR THE PURPOSE OF IDENTIFICATION IN ERROR PRINTOUTS. TEST 0 REFERS TO THE OPERATOR INPUT DIALOGUE ROUTINES DESCRIBED IN SECTIONS 8.1-8.3. THE OTHER TEST NUMBERS ARE ASSIGNED BELOW, IN THE DESCRIPTION OF PASS 1 AND PASS 2 TESTING.

IN THE FOLLOWING SECTIONS, TABLES A-G ARE REFERRED TO. IN THESE TABLES, DRIVES ARE NAMED FROM 0-17 FOR ILLUSTRATIVE PURPOSES, ALTHOUGH THE DRIVES ARE NAMED THE FOLLOWING WAY IN AN ACTUAL SITUATION : A0,A1, A2,...B0,B1,B2,...C0,C1,C2,... ETC. (SEE SECTION 8.0).

9.1 DESCRIPTION OF PASS 1 TESTS

IN PASS 1, THE BASIC READ/WRITE CAPABILITY OF EACH DRIVE IS DEMONSTRATED, AND COMPATIBILITY DATA PATTERNS ARE WRITTEN BY ALL DRIVES UPON THE SAME TEST CARTRIDGE.

THE SEQUENCE OF OPERATIONS PERFORMED ON EACH DRIVE IS AS FOLLOWS:

1. TEST 1 - MOUNTING OF TEST CARTRIDGE FOR PASS 1 - THE OPERATOR MOUNTS THE PACK ON THIS DRIVE AND MANUALLY LOADS THE HEADS, AS DIRECTED BY THE PROGRAM (SEE SECTION 8.4).

2. TEST 2 - BASIC READ/WRITE DATA TEST - THE PROGRAM PERFORMS A WRITE AND WRITE CHECK OPERATION USING A "WORST CASE" DATA PATTERN, AT THE APPROPRIATE SECTOR FOR THIS DRIVE (SEE TABLE A) ON ALL SURFACES. THE PURPOSE OF THIS OPERATION IS TO VERIFY THE BASIC READ/WRITE CAPABILITY OF THE DRIVE ON PASS 1. THE ENTIRE SECTOR IS WRITTEN WITH THE REPETITION OF THE DATA PATTERN SHOWN IN TABLE B. THIS PATTERN CONSISTS OF A MIXTURE OF HIGH AND LOW FREQUENCY ELEMENTS AND BIT STRINGS REQUIRING VARIOUS DEGREES OF PRECOMPENSATION, WHEN ENCODED.

3. TEST 3 - WRITE OVERWRITE AND COMPATIBILITY CYLINDER BLOCKS - NEXT, THE PROGRAM WRITES ALL SECTORS FOR THIS DRIVE WITHIN THE CYLINDER BLOCKS SHOWN IN TABLE C ON ALL SURFACES USING A SINGLE REPEATED WORD OF THE PATTERN IN TABLE G. LOGICAL DRIVE 0 USES WORD 0, LOGICAL DRIVE 1 USES WORD 1, LOGICAL DRIVE 10 USES WORD 10, ETC. THUS, THE DATA FROM EACH DRIVE IS UNIQUE. TABLE C HAS BEEN DETERMINED AS FOLLOWS:

   IN EACH OF THE SEVEN WRITE CURRENT ZONES ON EACH SURFACE SECTORS ARE WRITTEN WITHIN TWO DISTINCT CYLINDER BLOCKS. IN THE FIRST 16 CYLINDERS OF EACH WRITE CURRENT ZONE DATA IS WRITTEN TO BE LATER OVERWRITTEN IN PASS 2, DURING THE OVERWRITE TEST. IN THE LAST 16 CYLINDERS OF EACH CURRENT ZONE (EXCEPT THE INNERMOST ZONE) COMPATIBILITY DATA IS WRITTEN TO BE READ WITH OFFSET IN PASS 2. WITHIN EACH CURRENT ZONE BOTH THE OVERWRITE AND COMPATIBILITY CYLINDER BLOCKS ARE IDENTICALLY WRITTEN BUT FROM ZONE TO ZONE THE DATA WRITTEN BY EACH DRIVE IS ROTATED SEVERAL SECTORS SO THAT OVER ALL THE ZONES THE DATA APPEARS AT VARIOUS ANGULAR POSITIONS ON THE PACK.

   WITHIN EACH CYLINDER BLOCK, UP TO 16 SECTORS ARE WRITTEN (DEPENDING ON THE NUMBER OF DRIVES BEING TESTED) ON EACH CYLINDER. THESE SECTORS ARE ALWAYS SECTORS 0,1,2,3,5,6,7,10,12,13,14,15,17,20,21,22(OCTAL). OF THE REMAINING SECTORS, 4,11,16, AND 23 ARE USED FOR DRIVE SELF-TESTING IN PASS 2 (SEE SECTION 9.2).

   THE BASIC LAYOUT OF A TYPICAL CYLINDER BLOCK IS SHOWN IN TABLE D, WHERE THE BLOCK SHOWN IS THE COMPATIBILITY BLOCK FOR ZONE 1, WHICH STARTS ON CYLINDER 60, AND HAS THE ROTATING STARTING SECTOR = SECTOR 0. EACH NUMBER INSIDE THE BLOCK IS THE NUMBER OF THE DRIVE WHICH WRITES THAT SECTOR. TABLE D SHOWS THE BLOCKS WRITTEN BY EACH OF 16 DRIVES. IF ANY OF THE DRIVES SHOWN ARE NOT PRESENT, HOWEVER, THE BLOCKS RESERVED FOR THE MISSING DRIVES ARE SIMPLY NOT WRITTEN.

   THE ABOVE PATTERN OF SECTOR WRITES INSURES THAT DATA FROM EACH DRIVE IS WRITTEN ON ADJACENT CYLINDERS TO DATA FROM EVERY OTHER DRIVE, IN BOTH DIRECTIONS. IN ADDITION, THE ROTATION OF THE ABOVE SECTORS FROM CURRENT ZONE TO CURRENT ZONE INSURES THAT OVERWRITE AND DATA COMPATIBILITY TESTING IS DONE AT SEVERAL DIFFERENT ANGULAR POSITIONS WITH RESPECT TO THE CARTRIDGE.

4. TEST 4 - DISMOUNTING OF TEST CARTRIDGE IN PASS 1 - THE OPERATOR UNLOADS THE DRIVE AND DISMOUNTS THE PACK, AS DIRECTED BY THE PROGRAM (SEE SECTION 8.4), TO PROCEED WITH THE ABOVE STEPS ON THE NEXT DRIVE.

9.2 DESCRIPTION OF PASS 2 TESTS

IN PASS 2,THE ABILITY OF EACH DRIVE TO COMPLETELY OVERWRITE DATA WRITTEN BY ALL OTHER DRIVES AND TO READ DATA WRITTEN BY ALL OTHER DRIVES, IS TESTED.

THE SEQUENCE OF OPERATIONS PERFORMED BY EACH DRIVE IS AS FOLLOWS:

1. TEST 5 - MOUNTING OF TEST CARTRIDGE FOR PASS 2 - THE OPERATOR MOUNTS THE PACK ON THIS DRIVE AND MANUALLY LOADS THE HEADS, AS DIRECTED BY THE PROGRAM (SEE SECTION 8.5).

2. TEST 6 - OVERWRITE TEST - NEXT, THE PROGRAM PROCEEDS TO TEST THIS DRIVE'S OVERWRITE CAPABILITY. FIRST, THE APPROPRIATE CYLINDERS IN TABLE E FOR THIS DRIVE ARE OVERWRITTEN, ON EACH SURFACE. THE DATA USED IS A REPETITION OF A SINGLE WORD OF THE PATTERN IN TABLE G, LOGICAL DRIVE 0 USES WORD 0, LOGICAL DRIVE 1 USES WORD 1, LOGICAL DRIVE 10 USES WORD 10, ETC.

   THEN, EACH CYLINDER OVERWRITTEN IS READ BACK BY THIS DRIVE WITH THE FOLLOWING RANGE OF OFFSET VALUES : 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200 MICRO-INCHES, IN EACH OFFSET DIRECTION (+ AND -). THE PROGRAM SCANS FOR READ ERRORS (DCK,HVRC,ETC.) DURING THIS READ, AND IF ONE OCCURS, THE PROGRAM DETERMINES WHICH DRIVE'S DATA HAS NOT BEEN CORRECTLY OVERWRITTEN, AND A SCORE FOR THAT DRIVE IS DECREMENTED. THEN, THE TRANSFER IS CONTINUED AT THE NEXT SECTOR, WITH THAT OFFSET VALUE. THE READS ARE DONE WITH ALL OF THE ABOVE OFFSETS APPLIED, AND A SEPARATE SCORE FOR EACH DRIVE IS KEPT, WHILE THE CURRENT DRIVE IS PERFORMING THE OVERWRITES. FOR EACH TRACK (0,1,2), SCORES ARE AVERAGED OVER ALL CYLS TESTED, IN EACH OFFSET DIRECTION. AT THE COMPLETION OF THE OVERWRITE TEST ON THIS DRIVE, THE SCORES OF ALL THE DRIVES ARE CONVERTED AND STORED, FOR PRINTING AT THE END OF PASS 2 (AS DESCRIBED IN SECTION 10.2). EACH SCORE IS PROPORTIONAL TO THE DEGREE OF OFFSET WHICH COULD BE APPLIED IN A GIVEN OFFSET DIRECTION BY THE CURRENT DRIVE WHILE SUCCESSFULLY READING THE DATA IT WROTE OVER ONE OF THE OTHER DRIVE'S DATA. THUS, THE PRINTOUT REVEALS WHICH DRIVES ARE INVOLVED, IN A SITUATION IN WHICH A DRIVE CANNOT OVERWRITE ONE OR SEVERAL OTHER DRIVE'S DATA.

3. TEST 7 - DRIVE SELF-TEST - THE PROGRAM NEXT EVALUATES THE DRIVE'S ABILITY TO WRITE AND READ ITS OWN DATA, AT VARIOUS POSITIONS ON THE PACK. FIRST, SECTORS 4,11,16, AND 23 OF THE APPROPRIATE CYLINDERS SHOWN IN TABLE F FOR THIS DRIVE ARE WRITTEN WITH THE DATA PATTERN SHOWN IN TABLE B, FOR ALL SURFACES. THEN, THE SECTORS ARE READ WITH THE FOLLOWING OFFSETS APPLIED : 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200 MICRO-INCHES, IN EACH OFFSET DIRECTION. THE PROGRAM SCANS FOR READ ERRORS DURING EACH READ, AND IT COMPUTES A SCORE WHICH IS PROPORTIONAL TO THE FAILING OFFSET

MAGNITUDE. THEN, THE SCORES FOR ALL 4 SECTORS READ IN THIS CYLINDER BLOCK ARE AVERAGED, TO COME UP WITH A DRIVE SELF-TEST SCORE FOR EACH SURFACE FOR EACH OFFSET DIRECTION. THIS SCORE IS SAVED FOR LATER USE, TO BECOME THE STANDARD FOR THE COMPATIBILITY DATA READS WHICH ARE TO FOLLOW.

4. TEST 10 - COMPATIBILITY DATA READ TEST - HAVING ESTABLISHED A SELF-TEST SCORE FOR THIS DRIVE, THE PROGRAM PROCEEDS TO PERFORM THE COMPATIBILITY DATA READS OF THE PATTERNS WRITTEN BY ALL THE DRIVES IN EACH CYLINDER BLOCK (ON EACH SURFACE). EACH COMPATIBILITY CYLINDER BLOCK SHOWN IN TABLE C IS READ, A CYLINDER AT A TIME, FOR THE FOLLOWING OFFSET VALUES : 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200 MICRO-INCHES, IN EACH OFFSET DIRECTION. THE PROGRAM SCANS FOR READ ERRORS DURING EACH READ AND IF ONE OCCURS, THE PROGRAM DETERMINES WHICH DRIVE'S DATA WAS BEING READ AT THAT INSTANT AND A SCORE FOR THAT DRIVE IS DECREMENTED. THEN, THE TRANSFER IS CONTINUED AT THE NEXT SECTOR, WITH THAT OFFSET VALUE. THE READS ARE DONE WITH ALL OF THE ABOVE OFFSETS APPLIED, AND A SEPARATE SCORE FOR EACH DRIVE IS KEPT, WHILE THE CURRENT DRIVE IS READING THE COMPATIBILITY DATA. THEN, EACH SCORE IS APPROPRIATELY ADJUSTED TO REFLECT THE SELF-TEST SCORE FOR THE CURRENT DRIVE AT THAT PARTICULAR CYLINDER BLOCK. THE SCORES ARE THEN AVERAGED OVER ALL CYLINDER BLOCKS. EACH SCORE IS PROPORTIONAL TO THE CAPABILITY OF THE CURRENT DRIVE TO SUCCESSFULLY READ THE DATA WRITTEN BY ONE OF THE OTHER DRIVES, AND SCORES ARE COMPUTED SEPARATELY FOR EACH SURFACE (TRACK), FOR EACH OFFSET DIRECTION. THUS, THE PRINTOUT REVEALS WHICH DRIVES ARE INVOLVED IN A SITUATION IN WHICH A PARTICULAR DRIVE HAS DIFFICULTY IN READING THE DATA OF ONE OR SEVERAL OTHER DRIVES.

5. TEST 11 - TYPE TEST SCORES AND DISMOUNT PACK IN PASS 2 - THE OVERWRITE AND COMPATIBILITY DATA READ TEST SCORES FOR THIS DRIVE ARE CONVERTED AND TYPED. THEN, THE OPERATOR UNLOADS THE DRIVE AND DISMOUNTS THE PACK AS DIRECTED BY THE PROGRAM (SEE SECTION 8.5), TO PROCEED WITH THE ABOVE STEPS ON THE NEXT DRIVE.

10.0 PRINTOUT OF TEST RESULTS

THE TEST RESULTS ARE PRINTED AT THE END OF PASS 2 ON EACH DRIVE BEING TESTED. THESE RESULTS PERTAIN TO THE OVERWRITE TEST AND THE COMPATIBILITY DATA READ TEST.

10.1 OVERWRITE AND DRIVE COMPATIBILITY DATA TEST RESULTS

THE RESULTS OF BOTH THE OVERWRITE AND THE COMPATIBILITY DATA READ TESTS ARE ALWAYS PRINTED, REGARDLESS OF DEGREE OF SUCCESS. COMPLETE RESULTS ARE PRINTED, SEPARATELY FOR EACH DRIVE, AT THE END OF PASS 2 ON THAT DRIVE. THE TEST RESULTS ARE TABULAR IN FORM, AND CONSIST OF READ SCORES FOR THE CURRENT DRIVE, USING DATA WRITTEN BY ALL OTHER DRIVES. THE SCORES ARE CONVERTED ON THE BASIS OF 12(DEC), (WITH A PERFECT SCORE = 12). ALSO, THE SCORES ARE LISTED SEPARATELY FOR EACH TRACK (SURFACE), FOR EACH OFFSET DIRECTION, AND AN OVERALL AVERAGE FOR ALL THREE TRACKS, ON THE CURRENT DRIVE, IS PRINTED.

IN THE FOLLOWING EXAMPLE, THERE ARE 2 SYSTEMS, AND THE DRIVES BEING TESTED ARE A0,A1,A2,B0, AND B5. THE TEST RESULTS FOR DRIVE A1 ARE SHOWN BELOW:

SCORES FOR DRIVE A1 (0-12 DEC.) :

TRACK NO.	DRIVE READ	OVRWRT SCORE OFST-	OVRWRT SCORE OFST+	READ SCORE OFST-	READ SCORE OFST+
0	A0	12	12	12	12
0	SELF	12	12	12	12
0	A2	* 7	* 6	12	12
0	B0	11	12	11	11
0	B5	9	12	11	11
1	A0	12	11	12	11
1	SELF	11	10	12	12
1	A2	12	11	11	12
1	B0	12	12	9	9
1	B5	12	12	11	12
2	A0	12	12	12	12
2	SELF	12	12	12	12
2	A2	11	9	11	11
2	B0	12	12	12	12
2	B5	12	12	12	12

DRIVE A1 OVWRT AVG = 12
DRIVE A1 READ AVG = 10

THE ABOVE EXAMPLE REVEALS A POSSIBLE COMPATIBILITY PROBLEM BETWEEN DRIVES A1 AND A2. NOTICE THAT ON TRACK 0, THAT THE OVERWRITE SCORES

WERE UNACCEPTABLY LOW (7 OR LESS), AND THE PROGRAM NOTED THESE BAD SCORES WITH AN ASTERISK (*).

11.0 ERROR REPORTING

11.1 COMMON ERRORS

THE FOLLOWING IS A LIST OF COMMON ERROR MESSAGES WHICH ACCOMPANY ERROR TYPEOUTS FROM THE RK06 DRIVE COMPATIBILITY PROGRAM. THE ERRORS ARE SELF-EXPLANATORY.

1. UNIBUS PARITY ERROR
2. NON-EXISTANT MEMORY ERROR
3. NON-EXISTANT DRIVE ERROR
4. UNIT FIELD ERROR
5. SUBSYSTEM TIMEOUT
6. SERCON PARITY ERROR
7. DRIVE DETECTED PARITY ERROR
8. AC LOW
9. SPEED LOSS
10. ILLEGAL FUNCTION ERROR
11. PROGRAMMING ERROR
12. NON-EXECUTABLE FUNCTION ERROR
13. DRIVE TYPE ERROR
14. FORMAT ERROR
15. WRITE LOCK ERROR
16. DRIVE UNSAFE ERROR
17. SEEK INCOMPLETE ERROR
18. CYLINDER OVERFLOW ERROR
19. ILLEGAL CYLINDER ADDRESS ERROR

20. DRIVE OFF TRACK

21. DRIVE TIMING ERROR

22. DATA LATE ERROR

23. CONTROLLER TIMEOUT ERROR

24. OPERATION INCOMPLETE ERROR

25. HEADER VRC ERROR

26. DATA CHECK ERROR

27. WRITE CHECK ERROR

28. DATA MISCOMPARE

29. NO DRIVE RESPONSE - UFE AND NXD

30. DRIVE ERROR WILL NOT CLEAR

31. DRIVE STATUS CHANGE WILL NOT CLEAR

32. ATTENTION BUT NO STATUS CHANGE OR FAULT

33. ATTENTION BUT DRIVE NOT AVAILABLE

34. ERROR WHILE GATHERING DRIVE STATUS

35. MULTIPLE DRIVE SELECT

36. HEADER COMPARE ERROR

37. ERROR IN RECALIBRATE FOR RECOVERY

38. CLEAR CONTROLLER DID NOT CLEAR ERROR

39. NO ATTENTION IN ATTENTION SUMMARY REGISTER

40. UNSOLICITED ATTENTION

41. UNEXPECTED DATA TYPE ERROR

42. ATTENTION DID NOT RESET WITH CLEAR

43. SUBSYSTEM CLEAR DID NOT CLEAR DRIVE ATTENTION

44. DATA LATE WHEN UNLOADING HEADER

45. CONTROLLER ERROR WHEN DRIVER SERVICING

46. RETRY UNSUCCESSFUL

47. BAD SECTOR ERROR ON SECTOR NOT LISTED BAD

11.2 ERROR HANDLING

ERRORS REPORTED BY THE PROGRAM CONSIST OF COMMON FAILURES RESULTING FROM ATTEMPTED SUBSYSTEM FUNCTIONS, AS WELL AS CERTAIN ERRORS UNIQUE TO PARTICULAR TESTS. EACH ERROR PRINTOUT CONSISTS OF AN ERROR DESCRIPTION AND TEST NUMBER, POSSIBLY FOLLOWED BY HEADER LINES, COLUMN HEADINGS, AND COLUMNS OF REGISTER CONTENTS IN OCTAL. AS MUCH MEANINGFUL REGISTER DATA AS POSSIBLE (FOR EXAMPLE, RK611 REGISTERS) ARE REPORTED IN A GIVEN ERROR. OTHER ERROR REPORTS MAY CONSIST OF A SINGLE DESCRIPTIVE LINE.

11.3 ERROR PRINTOUT EXAMPLE

```
     ** WRITE CHECK ERROR

     PREVIOUS COMMAND:
     DRIVE     CMND     CYLNDR     TRACK     SECTOR     WD CNT
     000000    000121   000016     000001    000000     175000
       HI BA     LO BA
       000000    061566

     CURRENT COMMAND:
     ERR PC     DRIVE    CMND     CYLNDR     TRACK     SECTOR     WD CNT
     041154     000000   000131   000016     000001    000006     175000
       HI BA     LO BA
       000000    061566

     PACK ADDRESS OF ERROR(S):
     CYLNDR     TRACK     SECTOR
     000016     000001    000006
```

TABLE A

BASIC READ/WRITE TEST SECTORS

DRIVE NO.	ADDRESS OF SECTOR ON EACH SURFACE (IN OCTAL)	
	CYLINDER	SECTOR
	RK06/RK07	
0	620/1444	0
1	620/1444	2
2	620/1444	4
3	620/1444	6
4	620/1444	10
5	620/1444	12
6	620/1444	14
7	620/1444	16
10	622/1446	0
11	622/1446	2
12	622/1446	4
13	622/1446	6
14	622/1446	10
15	622/1446	12
16	622/1446	14
17	622/1446	16

1059
1060
1061
1062
1063
1064
1065
1066
1067

TABLE B

1068
1069
1070

WORST CASE DATA PATTERN (REPEATS EVERY 16 WORDS)

1071
1072
1073
1074
1075
1076
1077

WORD NO.	DATA (OCTAL)
0	072307
1	135143
2	156461
3	167230
4	073514
5	035646
6	016723
7	107351
10	143564
11	061672
12	030735
13	114356
14	046167
15	123073
16	151453
17	164616

1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093

TABLE C

CYLINDER BLOCK ASSIGNMENT FOR A GIVEN SURFACE

CURRENT ZONE - RANGE	OVERWRITE CYL BLK RANGE (OCT)	COMPATIBILITY CYL BLK RANGE (OCT)	ROTATING STARTING SECTOR	
1 - CYL 0-77	CYL 0-17	CYL 60-77	SECT 0	RK06
2 - 100-177	100-117	160-177	3	RK06
3 - 200-277	200-217	260-277	7	RK06
4 - 300-377	300-317	360-377	13	RK06
5 - 400-477	400-417	460-477	17	RK06
6 - 500-577	500-517	560-577	22	RK06
7 - 600-632	600-617	---	0	RK06
1 - CYL 0-177	CYL 0-17	CYL 160-177	SECT 0	RK07
2 - 200-377	200-217	360-377	3	RK07
3 - 400-577	400-417	560-577	7	RK07
4 - 600-777	600-617	760-777	13	RK07
5 - 1000-1177	1000-1017	1160-1177	17	RK07
6 - 1200-1377	1200-1217	1360-1377	22	RK07
7 - 1400-1456	1400-1417	---	0	RK07

TABLE D

BASIC CYLINDER BLOCK LAYOUT EXAMPLE

(SHOWN FOR RK06, USE CYL 160-177 FOR RK07)

CYLINDER NUMBERS (OCTAL)	SECTOR NUMBERS (OCTAL)															
	0	1	2	3	5	6	7	10	12	13	14	15	17	20	21	22
60	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
61	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	0
62	3	4	5	6	7	10	11	12	13	14	15	16	17	0	1	2
63	6	7	10	11	12	13	14	15	16	17	0	1	2	3	4	5
64	12	13	14	15	16	17	0	1	2	3	4	5	6	7	10	11
65	17	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16
66	5	6	7	10	11	12	13	14	15	16	17	0	1	2	3	4
67	14	15	16	17	0	1	2	3	4	5	6	7	10	11	12	13
70	4	5	6	7	10	11	12	13	14	15	16	17	0	1	2	3
71	15	16	17	0	1	2	3	4	5	6	7	10	11	12	13	14
72	7	10	11	12	13	14	15	16	17	0	1	2	3	4	5	6
73	2	3	4	5	6	7	10	11	12	13	14	15	16	17	0	1
74	16	17	0	1	2	3	4	5	6	7	10	11	12	13	14	15
75	13	14	15	16	17	0	1	2	3	4	5	6	7	10	11	12
76	11	12	13	14	15	16	17	0	1	2	3	4	5	6	7	10
77	10	11	12	13	14	15	16	17	0	1	2	3	4	5	6	7

TABLE E

OVERWRITE CYLINDERS

DRIVE #	CYLINDERS OVERWRITTEN (OCTAL)
0	0,100,200,300,400,500,600
1	1,101,201,301,401,501,601
2	2,102,202,302,402,502,602
3	3,103,203,303,403,503,603
4	4,104,204,304,404,504,604
5	5,105,205,305,405,505,605
6	6,106,206,306,406,506,606
7	7,107,207,307,407,507,607
10	10,110,210,310,410,510,610
11	11,111,211,311,411,511,611
12	12,112,212,312,412,512,612
13	13,113,213,313,413,513,613
14	14,114,214,314,414,514,614
15	15,115,215,315,415,515,615
16	16,116,216,316,416,516,616
17	17,117,217,317,417,517,617

NOTE: THE ,BOVE TABLE GIVES CYLINDER NUMBERS FOR THE RK06.
RE'ER FOR TABLE C TO OBTAIN CORRESPONDING CYLINDER
NUMBERS FOR AN RK07.

1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248

TABLE F

SELF-TEST CYLINDERS

DRIVE #	CYLINDERS (WHERE SECTORS 4,11,16,23 ARE TESTED)
0	60,160,260,360,460,560
1	61,161,261,361,461,561
2	62,162,262,362,462,562
3	63,163,263,363,463,563
4	64,164,264,364,464,564
5	65,165,265,365,465,565
6	66,166,266,366,466,566
7	67,167,267,367,467,567
10	70,170,270,370,470,570
11	71,171,271,371,471,571
12	72,172,272,372,472,572
13	73,173,273,373,473,573
14	74,174,274,374,474,574
15	75,175,275,375,475,575
16	76,176,276,376,476,576
17	77,177,277,377,477,577

NOTE: THE ABOVE TABLE GIVES CYLNDER NULMBERS FOR THE RK06.
REFER TO TABLE C TO OBTAIN CORRESPONDING CYLINDER
NUMBERS FOR THE RK07.

TABLE G

PSEUDO-RANDOM DATA PATTERN

WORD #	DATA (OCTAL)
0	040135
1	177070
2	070414
3	064531
4	174473
5	062422
6	114352
7	036620
10	010031
11	052336
12	017310
13	011347
14	102367
15	152567
16	001246
17	160073

```
                                        .NLIST MC,MD,CND
                                        .LIST   ME
                                        .ENABL  ABS,AMA

                                ;*** REV 003 ***




                                ;;*****************************************************************
                                .SBTTL  STARTING ADDRESSES
                                ;*
                                ;*      200     DEFAULT PARAMETERS AND RUN PASS 1 AND PASS 2
                                ;*      204     SELECT PARAMETERS AND RUN PASS 1
                                ;*      220     SELECT PARAMETERS AND RUN PASS 2
                                ;*      224     MEMORY DUMP ROUTINE
                                ;;*****************************************************************


000001                          $TN=1
                                .TITLE  CZR6QC0 RK6 DR CPT PROG
                                ;*COPYRIGHT (C) 1977,1980
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
160000                          $SWR=160000     ;;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT

163000                                  $SWR=163000
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        12            REPORT DESCRIPTION ONLY, ON ERRORS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            APPLY RANDOM STALL BETWEEN OPERATIONS
                                ;*         7            TYPE BAD SECTOR FILES (BSF'S) AT START

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
```

```
 1341          000012                  LF=     12              ;;CODE FOR LINE FEED
 1342          000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
 1343          000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
 1344          177776                  PS=     177776          ;;PROCESSOR STATUS WORD
 1345                                  .EQUIV  PS,PSW
 1346          177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
 1347          177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
 1348          177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
 1349          177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
 1350
 1351                                  ;*GENERAL PURPOSE REGISTER DEFINITIONS
 1352          000000                  R0=     %0              ;;GENERAL REGISTER
 1353          000001                  R1=     %1              ;;GENERAL REGISTER
 1354          000002                  R2=     %2              ;;GENERAL REGISTER
 1355          000003                  R3=     %3              ;;GENERAL REGISTER
 1356          000004                  R4=     %4              ;;GENERAL REGISTER
 1357          000005                  R5=     %5              ;;GENERAL REGISTER
 1358          000006                  R6=     %6              ;;GENERAL REGISTER
 1359          000007                  R7=     %7              ;;GENERAL REGISTER
 1360          000006                  SP=     %6              ;;STACK POINTER
 1361          000007                  PC=     %7              ;;PROGRAM COUNTER
 1362
 1363                                  ;*PRIORITY LEVEL DEFINITIONS
 1364          000000                  PR0=    0               ;;PRIORITY LEVEL 0
 1365          000040                  PR1=    40              ;;PRIORITY LEVEL 1
 1366          000100                  PR2=    100             ;;PRIORITY LEVEL 2
 1367          000140                  PR3=    140             ;;PRIORITY LEVEL 3
 1368          000200                  PR4=    200             ;;PRIORITY LEVEL 4
 1369          000240                  PR5=    240             ;;PRIORITY LEVEL 5
 1370          000300                  PR6=    300             ;;PRIORITY LEVEL 6
 1371          000340                  PR7=    340             ;;PRIORITY LEVEL 7
 1372
 1373                                  ;*"SWITCH REGISTER" SWITCH DEFINITIONS
 1374          100000                  SW15=   100000
 1375          040000                  SW14=   40000
 1376          020000                  SW13=   20000
 1377          010000                  SW12=   10000
 1378          004000                  SW11=   4000
 1379          002000                  SW10=   2000
 1380          001000                  SW09=   1000
 1381          000400                  SW08=   400
 1382          000200                  SW07=   200
 1383          000100                  SW06=   100
 1384          000040                  SW05=   40
 1385          000020                  SW04=   20
 1386          000010                  SW03=   10
 1387          000004                  SW02=   4
 1388          000002                  SW01=   2
 1389          000001                  SW00=   1
 1390                                  .EQUIV  SW09,SW9
 1391                                  .EQUIV  SW08,SW8
 1392                                  .EQUIV  SW07,SW7
 1393                                  .EQUIV  SW06,SW6
 1394                                  .EQUIV  SW05,SW5
 1395                                  .EQUIV  SW04,SW4
 1396                                  .EQUIV  SW03,SW3
```

```
                                       .EQUIV  SW02,SW2
                                       .EQUIV  SW01,SW1
                                       .EQUIV  SW00,SW0

                                       ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                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
                                       .EQUIV  BIT09,BIT9
                                       .EQUIV  BIT08,BIT8
                                       .EQUIV  BIT07,BIT7
                                       .EQUIV  BIT06,BIT6
                                       .EQUIV  BIT05,BIT5
                                       .EQUIV  BIT04,BIT4
                                       .EQUIV  BIT03,BIT3
                                       .EQUIV  BIT02,BIT2
                                       .EQUIV  BIT01,BIT1
                                       .EQUIV  BIT00,BIT0

                                       ;*BASIC "CPU" TRAP VECTOR ADDRESSES
             000004                    ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
             000010                    RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
             000014                    TBITVEC=14              ;;"T" BIT
             000014                    TRTVEC= 14              ;;TRACE TRAP
             000014                    BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
             000020                    IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
             000024                    PWRVEC= 24              ;;POWER FAIL
             000030                    EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
             000034                    TRAPVEC=34              ;;"TRAP" TRAP
             000060                    TKVEC=  60              ;;TTY KEYBOARD VECTOR
             000064                    TPVEC=  64              ;;TTY PRINTER VECTOR
             000240                    PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

                                       .SBTTL  TRAP CATCHER

             000000                            .=0
                                       ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                       ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                       ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
             000174                            .=174
000174  000000                       DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                       SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                       .SBTTL  STARTING ADDRESS(ES)
```

```
 1453  000200  000137  012146                  JMP     @#DFSTRT        ;;JUMP TO STARTING ADDRESS OF PROGRAM
 1454          000204                          .=204
 1455  000204  000137  012154                  JMP     @#P1STRT
 1456          000220                          .=220
 1457  000220  000137  012172                  JMP     @#P2STRT
 1458
 1459  000224  000700                  ..LOW:  .WORD   700
 1460  000226  001100                  ..HIGH: .WORD   1100
 1461
 1462                                  .SBTTL  ACT11 HOOKS
 1463
 1464                                  ;;**************************************************************
 1465                                  ;HOOKS REQUIRED BY ACT11
 1466          000230                          $SVPC=.                 ;SAVE PC
 1467          000046                          .=46
 1468  000046  017234                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
 1469          000052                          .=52
 1470  000052  140000                          .WORD   140000          ;;2)SET LOC.52 TO 140000
 1471          000230                          .=$SVPC                 ;; RESTORE PC
 1472          001000                          .=1000
 1473                                  .SBTTL  APT PARAMETER BLOCK
 1474
 1475                                  ;;**************************************************************
 1476                                  ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
 1477                                  ;;**************************************************************
 1478          001000                          .$X=.   ;;SAVE CURRENT LOCATION
 1479          000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
 1480  000024  000200                          200     ;;FOR APT START UP
 1481          000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
 1482  000044  001000                          $APTHDR ;;POINT TO APT HEADER BLOCK
 1483          001000                          .=.$X   ;;RESET LOCATION COUNTER
 1484                                  ;;**************************************************************
 1485                                  ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
 1486                                  ;INTERFACE SPEC.
 1487
 1488  001000                          $APTHD:
 1489  001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
 1490  001002  001330                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
 1491  001004  001320                  $TSTM:  .WORD   1320    ;;RUN TIM OF LONGEST TEST
 1492  001006  001546                  $PASTM: .WORD   1546    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
 1493  001010  001546                  $UNITM: .WORD   1546    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
 1494  001012  000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
 1495          120210                  AVECT1=120210                   ;RKVEC=210, RKPRI=5
 1496          177440                  ABASE=177440                    ;RKBAS ADRS
 1497          000377                  ADEVM=000377                    ;SET DEVICES 0-7 IN MAP
```

```
                                        .SBTTL  COMMON TAGS

                                        ;;**********************************************************************
                                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                        ;*USED IN THE PROGRAM.

                001100                          .=1100
001100                          $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                          .WORD   0
001102     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000          $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
                                                        ;;WHICH  ($REG0) WAS OBTAINED
001162  000000          $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
001164  000000          $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
001166  000000          $REG2:  .WORD   0               ;;CONTAINS (($REGAD)+4)
001170  000000          $REG3:  .WORD   0               ;;CONTAINS (($REGAD)+6)
001172  000000          $REG4:  .WORD   0               ;;CONTAINS (($REGAD)+10)
001174  000000          $REG5:  .WORD   0               ;;CONTAINS (($REGAD)+12)
001176  000000          $REG6:  .WORD   0               ;;CONTAINS (($REGAD)+14)
001200  000000          $REG7:  .WORD   0               ;;CONTAINS (($REGAD)+16)
001202  000000          $REG10: .WORD   0               ;;CONTAINS (($REGAD)+20)
001204  000000          $REG11: .WORD   0               ;;CONTAINS (($REGAD)+22)
001206  000000          $REG12: .WORD   0               ;;CONTAINS (($REGAD)+24)
001210  000000          $REG13: .WORD   0               ;;CONTAINS (($REGAD)+26)
001212  000000          $REG14: .WORD   0               ;;CONTAINS (($REGAD)+30)
001214  000000          $REG15: .WORD   0               ;;CONTAINS (($REGAD)+32)
001216  000000          $REG16: .WORD   0               ;;CONTAINS (($REGAD)+34)
001220  000000          $REG17: .WORD   0               ;;CONTAINS (($REGAD)+36)
001222  000000          $REG20: .WORD   0               ;;CONTAINS (($REGAD)+40)
```

```
 1554  001224  000000                  $REG21: .WORD   0               ;;CONTAINS (($REGAD)+42)
 1555  001226  000000                  $REG22: .WORD   0               ;;CONTAINS (($REGAD)+44)
 1556  001230  000000                  $REG23: .WORD   0               ;;CONTAINS (($REGAD)+46)
 1557  001232  000000                  $REG24: .WORD   0               ;;CONTAINS (($REGAD)+50)
 1558  001234  000000                  $REG25: .WORD   0               ;;CONTAINS (($REGAD)+52)
 1559  001236  000000                  $REG26: .WORD   0               ;;CONTAINS (($REGAD)+54)
 1560  001240  000000                  $REG27: .WORD   0               ;;CONTAINS (($REGAD)+56)
 1561  001242  000000                  $REG30: .WORD   0               ;;CONTAINS (($REGAD)+60)
 1562  001244  000000                  $REG31: .WORD   0               ;;CONTAINS (($REGAD)+62)
 1563  001246  000000                  $REG32: .WORD   0               ;;CONTAINS (($REGAD)+64)
 1564  001250  000000                  $REG33: .WORD   0               ;;CONTAINS (($REGAD)+66)
 1565  001252  000000                  $REG34: .WORD   0               ;;CONTAINS (($REGAD)+70)
 1566  001254  000000                  $REG35: .WORD   0               ;;CONTAINS (($REGAD)+72)
 1567  001256  000000                  $REG36: .WORD   0               ;;CONTAINS (($REGAD)+74)
 1568  001260  000000                  $REG37: .WORD   0               ;;CONTAINS (($REGAD)+76)
 1569  001262  000000                  $TMP0:  .WORD   0               ;;USER DEFINED
 1570  001264  000000                  $TMP1:  .WORD   0               ;;USER DEFINED
 1571  001266  000000                  $TMP2:  .WORD   0               ;;USER DEFINED
 1572  001270  000000                  $TMP3:  .WORD   0               ;;USER DEFINED
 1573  001272  000000                  $TMP4:  .WORD   0               ;;USER DEFINED
 1574  001274  000000                  $TMP5:  .WORD   0               ;;USER DEFINED
 1575  001276  000000                  $TMP6:  .WORD   0               ;;USER DEFINED
 1576  001300  000000                  $TMP7:  .WORD   0               ;;USER DEFINED
 1577  001302  000000                  $TMP10: .WORD   0               ;;USER DEFINED
 1578  001304  000000                  $TMP11: .WORD   0               ;;USER DEFINED
 1579  001306  000000                  $TMP12: .WORD   0               ;;USER DEFINED
 1580  001310  000000                  $TMP13: .WORD   0               ;;USER DEFINED
 1581  001312  000000                  $TMP14: .WORD   0               ;;USER DEFINED
 1582  001314  000000                  $TMP15: .WORD   0               ;;USER DEFINED
 1583  001316  000000                  $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
 1584  001320  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 1585  001324     077                  $QUES:  .ASCII  /?/             ;;QUESTION MARK
 1586  001325     015                  $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
 1587  001326  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
 1588                                  ;;**************************************************************
 1589                                  .SBTTL  APT MAILBOX-ETABLE
 1590
 1591                                  ;;**************************************************************
 1592                                  .EVEN
 1593  001330                          $MAIL:                          ;;APT MAILBOX
 1594  001330  000000                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
 1595  001332  000000                  $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
 1596  001334  000000                  $TESTN: .WORD   ATESTN   ;;TEST NUMBER
 1597  001336  000000                  $PASS:  .WORD   APASS    ;;PASS COUNT
 1598  001340  000000                  $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
 1599  001342  000000                  $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
 1600  001344  000000                  $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
 1601  001346  000000                  $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
 1602  001350                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
 1603  001350     000                  $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
 1604  001351     000                  $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
 1605  001352  000000                  $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
 1606  001354  000000                  $USWR:  .WORD   AUSWR    ;;USER SWITCHES
 1607  001356  000000                  $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
 1608                                  ;*                      BITS 15-11=CPU TYPE
 1609                                  ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
```

```
                                        ;*                          11/70=C6,PDQ=07,Q=10
                                        ;*                  BIT 10=REAL TIME CLOCK
                                        ;*                  BIT  9=FLOATING POINT PROCESSOR
                                        ;*                  BIT  8=MEMORY MANAGEMENT
001360      000                         $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001361      000                         $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                                        ;*                  MEM.TYPE BYTE   --  (HIGH BYTE)
                                        ;*                          900 NSEC CORE=001
                                        ;*                          300 NSEC BIPOLAR=002
                                        ;*                          500 NSEC MOS=003
001362  000000                          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                                        ;*                    MEM.LAST ADDP.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001364      000                         $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001365      000                         $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001366  000000                          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001370      000                         $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001371      000                         $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001372  000000                          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001374      000                         $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001375      000                         $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001376  000000                          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001400  120210                          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001402  000000                          $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001404  177440                          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001406  000377                          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001410                                  $ETEND:
                                        .MEXIT
```

```
                                        .SBTTL  ERROR POINTER TABLE

                                        ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                        ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                        ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                        ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                        ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                        ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                        ;*      DH              ;;POINTS TO THE DATA HEADER
                                        ;*      DT              ;;POINTS TO THE DATA
                                        ;*      DF              ;;POINTS TO THE DATA FORMAT


001410                                  $ERRTB:

                                                ;ERROR 1                ;UNIBUS PARITY ERROR
001410  045405                                  EM1
001412  050500                                  DH100
001414  052264                                  DT100
001416  052374                                  DF01

                                                ;ERROR 2                ;NON-EXISTANT MEMORY
001420  045431                                  EM2
001422  050500                                  DH100
001424  052264                                  DT100
001426  052374                                  DF01

                                                ;ERROR 3                ;NON-EXISTANT DRIVE
001430  045455                                  EM3
001432  050500                                  DH100
001434  052264                                  DT100
001436  052374                                  DF01

                                                ;ERROR 4                ;UNIT FIELD ERROR
001440  045500                                  EM4
001442  050500                                  DH100
001444  052264                                  DT100
001446  052374                                  DF01

                                                ;ERROR 5                ;SUBSYSTEM TIMEOUT
001450  045521                                  EM5
001452  050500                                  DH100
001454  052264                                  DT100
001456  052424                                  DF02

                                                ;ERROR 6
001460  045540                                  EM6                     ;D TO C PARITY ERROR
001462  050500                                  DH100
001464  052264                                  DT100
001466  052460                                  DF03

                                                ;ERROR 7
001470  045564                                  EM7                     ;DRIVE DETECTED PARITY ERROR
001472  050500                                  DH100
001474  052264                                  DT100
```

```
 1693  001476  052460                          DF03
 1694
 1695                                          ;ERROR 10
 1696  001500  045620                          EM10                    ;AC LOW
 1697  001502  050500                          DH100
 1698  001504  052264                          DT100
 1699  001506  052424                          DF02
 1700
 1701                                          ;ERROR 11
 1702  001510  045627                          EM11                    ;SPEED LOSS
 1703  001512  050500                          DH100
 1704  001514  052264                          DT100
 1705  001516  052424                          DF02
 1706
 1707                                          ;ERROR 12
 1708  001520  045642                          EM12                    ;ILLEGAL FUNCTION
 1709  001522  050500                          DH100
 1710  001524  052264                          DT100
 1711  001526  052424                          DF02
 1712
 1713                                          ;ERROR 13
 1714  001530  045663                          EM13                    ;PROGRAMMING ERROR
 1715  001532  050500                          DH100
 1716  001534  052264                          DT100
 1717  001536  052374                          DF01
 1718
 1719                                          ;ERROR 14               ;NON-EXISTANT FUNCTION
 1720  001540  045705                          EM14
 1721  001542  050500                          DH100
 1722  001544  052264                          DT100
 1723  001546  052424                          DF02
 1724
 1725                                          ;ERROR 15
 1726  001550  045733                          EM15                    ;DRIVE TYPE ERROR
 1727  001552  050500                          DH100
 1728  001554  052264                          DT100
 1729  001556  052424                          DF02
 1730
 1731                                          ;ERROR 16
 1732  001560  045754                          EM16                    ;FORMAT ERROR
 1733  001562  050500                          DH100
 1734  001564  052264                          DT100
 1735  001566  052424                          DF02
 1736
 1737                                          ;ERROR 17
 1738  001570  045771                          EM17                    ;WRITE LOCK ERROR
 1739  001572  050500                          DH100
 1740  001574  052264                          DT100
 1741  001576  052424                          DF02
 1742
 1743                                          ;ERROR 20
 1744  001600  046012                          EM20                    ;DRIVE UNSAFE
 1745  001602  050500                          DH100
 1746  001604  052264                          DT100
 1747  001606  052424                          DF02
 1748
```

```
 1749                                           ;ERROR 21
 1750  001610  046027                          EM21                    ;SEEK INCOMPLETE
 1751  001612  050500                          DH100
 1752  001614  052264                          DT100
 1753  001616  052424                          DF02
 1754
 1755                                           ;ERROR 22
 1756  001620  046047                          EM22                    ;CYLINDER OVERFLOW
 1757  001622  050500                          DH100
 1758  001624  052264                          DT100
 1759  001626  052424                          DF02
 1760
 1761                                           ;ERROR 23
 1762  001630  046071                          EM23                    ;ILLEGAL CYLINDER
 1763  001632  050500                          DH100
 1764  001634  052264                          DT100
 1765  001636  052424                          DF02
 1766
 1767                                           ;ERROR 24
 1768  001640  046122                          EM24                    ;DRIVE OFF TRACK
 1769  001642  050500                          DH100
 1770  001644  052264                          DT100
 1771  001646  052424                          DF02
 1772
 1773                                           ;ERROR 25
 1774  001650  046142                          EM25                    ;DRIVE TIMING ERROR
 1775  001652  050500                          DH100
 1776  001654  052264                          DT100
 1777  001656  052424                          DF02
 1778
 1779                                           ;ERROR 26
 1780  001660  046165                          EM26                    ;DATA LATE
 1781  001662  050500                          DH100
 1782  001664  052264                          DT100
 1783  001666  052424                          DF02
 1784
 1785                                           ;ERROR 27
 1786  001670  046177                          EM27                    ;CONTROLLER TIMEOUT
 1787  001672  050500                          DH100
 1788  001674  052264                          DT100
 1789  001676  052424                          DF02
 1790
 1791                                           ;ERROR 30               ;OPERATION INCOMPLETE
 1792  001700  046222                          EM30
 1793  001702  050500                          DH100
 1794  001704  052264                          DT100
 1795  001706  052534                          DF05
 1796
 1797                                           ;ERROR 31
 1798  001710  046247                          EM31                    ;HEADER VRC ERROR
 1799  001712  050500                          DH100
 1800  001714  052264                          DT100
 1801  001716  052534                          DF05
 1802
 1803                                           ;ERROR 32
 1804  001720  046270                          EM32                    ;DATA CHECK ERROR
```

```
 1805  001722  050500                        DH100
 1806  001724  052264                        DT100
 1807  001726  052570                        DF07
 1808
 1809                                        ;ERROR 33
 1810  001730  046311                        EM33                    ;WRITE CHECK ERROR
 1811  001732  050500                        DH100
 1812  001734  052264                        DT100
 1813  001736  052624                        DF10
 1814
 1815                                        ;ERROR 34
 1816  001740  046333                        EM34                    ;DATA MISCOMPARE(S)
 1817  001742  050500                        DH100
 1818  001744  052264                        DT100
 1819  001746  052520                        DF04
 1820
 1821                                        ;ERROR 35
 1822  001750  046353                        EM35                    ;NO DRIVE RESPONSE-UFE & NXD
 1823  001752  050500                        DH100
 1824  001754  052264                        DT100
 1825  001756  052374                        DF01
 1826
 1827                                        ;ERROR 36
 1828  001760  046407                        EM36                    ;DRIVE ERROR WILL NOT CLEAR
 1829  001762  000000                        0
 1830  001764  000000                        0
 1831  001766  000000                        0
 1832
 1833                                        ;ERROR 37               ;DRIVE STATUS CHANGE WILL NOT CLEAR
 1834  001770  046442                        EM37
 1835  001772  000000                        0
 1836  001774  000000                        0
 1837  001776  000000                        0
 1838
 1839                                        ;ERROR 40
 1840  002000  046505                        EM40                    ;ATTENTION BUT NO STATUS CHANGE OR FAULT
 1841  002002  050500                        DH100
 1842  002004  052264                        DT100
 1843  002006  052424                        DF02
 1844
 1845                                        ;ERROR 41
 1846  002010  046551                        EM41                    ;ATTENTION BUT DRIVE NOT AVAILABLE
 1847  002012  050500                        DH100
 1848  002014  052264                        DT100
 1849  002016  052424                        DF02
 1850
 1851                                        ;ERROR 42
 1852  002020  046607                        EM42                    ;ATTENTION WHEN NOT EXPECTED
 1853  002022  050500                        DH100
 1854  002024  052264                        DT100
 1855  002026  052424                        DF02
 1856
 1857                                        ;ERROR 43               ;ERROR WHILE GATHERING DRIVE STATUS
 1858  002030  046637                        EM43
 1859  002032  050500                        DH100
 1860  002034  052264                        DT100
```

```
1861  002036  052670                        DF12
1862
1863                                        ;ERROR 44
1864  002040  047114                        EM63            ;CLEAR CONTROLLER DID NOT CLEAR ERROR
1865  002042  050500                        DH100
1866  002044  052264                        DT100
1867  002046  052670                        DF12
1868
1869                                        ;ERROR 45
1870  002050  047161                        EM64            ;NO ATTENTION IN ATTENTION SUMMARY REG
1871  002052  050500                        DH100
1872  002054  052264                        DT100
1873  002056  052670                        DF12
1874
1875                                        ;ERROR 46
1876  002060  047224                        EM65            ;UNSOLICITED ATTENTION
1877  002062  050500                        DH100
1878  002064  052264                        DT100
1879  002066  052670                        DF12
1880
1881                                        ;ERROR 47
1882  002070  047252                        EM66            ;UNEXPECTED DATA TYPE ERROR
1883  002072  050500                        DH100
1884  002074  052264                        DT100
1885  002076  052670                        DF12
1886
1887                                        ;ERROR 50
1888  002100  047305                        EM67            ;ATTENTION DID NOT RESET WITH CLEAR
1889  002102  050500                        DH100
1890  002104  052264                        DT100
1891  002106  052670                        DF12
1892
1893                                        ;ERROR 51
1894  002110  047344                        EM70            ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE ATTENTION
1895  002112  050500                        DH100
1896  002114  052264                        DT100
1897  002116  052670                        DF12
1898
1899                                        ;ERROR 52
1900  002120  046674                        EM52            ;MULTIPLE DRIVE SELECT
1901  002122  050500                        DH100
1902  002124  052264                        DT100
1903  002126  052670                        DF12
1904
1905                                        ;ERROR 53
1906  002130  046722                        EM53            ;ABREVIATED HCE ERROR
1907  002132  050500                        DH100
1908  002134  052264                        DT100
1909  002136  052720                        DF13
1910
1911                                        ;ERROR 54
1912  002140  046222                        EM30            ;OPERATION INCOMPLETE ERROR
1913  002142  050500                        DH100
1914  002144  052264                        DT100
1915  002146  052750                        DF14
1916
```

```
1917                                    ;ERROR 55
1918  002150  046247                    EM31                ;ABREVIATED HVRC ERROR
1919  002152  050500                    DH100
1920  002154  052264                    DT100
1921  002156  052720                    DF13
1922
1923                                    ;ERROR 56
1924  002160  046747                    EM56                ;2 TIMEOUT ERROR
1925  002162  050500                    DH100
1926  002164  052264                    DT100
1927  002166  053000                    DF15
1928
1929                                    ;ERROR 57           ;2ND LEVEL IN SUBSYSTEM TIMEOUT
1930  002170  046747                    EM56
1931  002172  050500                    DH100
1932  002174  052264                    DT100
1933  002176  053044                    DF16
1934
1935                                    ;ERROR 60
1936  002200  046766                    EM60                ;ERROR IN RECAL FOR RECOVERY
1937  002202  000000                    0
1938  002204  000000                    0
1939  002206  000000                    0
1940
1941                                    ;ERROR 61
1942  002210  047022                    EM61                ;ABORT MESSAGE
1943  002212  000000                    0
1944  002214  000000                    0
1945  002216  000000                    0
1946
1947                                    ;ERROR 62
1948  002220  047070                    EM62                ;CYLINDER MISCOMPARE
1949  002222  050500                    DH100
1950  002224  052264                    DT100
1951  002226  053110                    DF17
1952
1953                                    ;ERROR 63           ;DATA ERROR WORDS
1954  002230  000000                    0
1955  002232  000000                    0
1956  002234  052356                    DT602
1957  002236  053220                    DF25
1958
1959                                    ;ERROR 64
1960  002240  047114                    EM63                ;CLEAR CONTROLLER DID NOT CLEAR ERROR
1961  002242  050500                    DH100
1962  002244  052264                    DT100
1963  002246  052424                    DF02
1964
1965                                    ;ERROR 65
1966  002250  047161                    EM64                ;NO ATTENTION IN ATTENTION SUMMARY REG
1967  002252  050500                    DH100
1968  002254  052264                    DT100
1969  002256  052424                    DF02
1970
1971                                    ;ERROR 66
1972  002260  047224                    EM65                ;UNSOLICITED ATTENTION
```

```
 1973  002262  050500                        DH100
 1974  002264  052264                        DT100
 1975  002266  052424                        DF02
 1976
 1977                                        ;ERROR 67
 1978  002270  047252                        EM66                ;UNEXPECTED DATA TYPE ERROR
 1979  002272  050500                        DH100
 1980  002274  052264                        DT100
 1981  002276  052424                        DF02
 1982
 1983                                        ;ERROR 70
 1984  002300  047305                        EM67                ;ATTENTION DID NOT RESET WITH CLEAR
 1985  002302  050500                        DH100
 1986  002304  052264                        DT100
 1987  002306  052424                        DF02
 1988
 1989                                        ;ERROR 71
 1990  002310  047344                        EM70                ;SUBSYSTEM CLEAR DID NOT CLEAR ATT
 1991  002312  050500                        DH100
 1992  002314  052264                        DT100
 1993  002316  052424                        DF02
 1994
 1995                                        ;ERROR 72
 1996  002320  047413                        EM71                ;DATA LATE WHEN UNLOADING HEADER
 1997  002322  050500                        DH100
 1998  002324  052264                        DT100
 1999  002326  052424                        DF02
 2000
 2001                                        ;ERROR 73
 2002  002330  047453                        EM72                ;CONTROLLER ERROR DURING DRIVER SERVICE
 2003  002332  050500                        DH100
 2004  002334  052264                        DT100
 2005  002336  052424                        DF02
 2006
 2007                                        ;ERROR 74
 2008  002340  047522                        EM73                ;DRIVE DETECTED PARITY ERROR
 2009  002342  050500                        DH100
 2010  002344  052264                        DT100
 2011  002346  052424                        DF02
 2012
 2013                                        ;ERROR 75
 2014  002350  047556                        EM74                ;UNDEFINED ERROR
 2015  002352  050500                        DH100
 2016  002354  052264                        DT100
 2017  002356  052424                        DF02
 2018
 2019                                        ;ERROR 76
 2020  002360  047576                        EM75                ;MARKING SECTOR BAD MESSAGE
 2021  002362  000000                        0
 2022  002364  000000                        0
 2023  002366  000000                        0
 2024
 2025                                        ;ERROR 77
 2026  002370  047626                        EM76                ;BAD DATA VERIFICATION WITH READ
 2027  002372  051465                        DH605
 2028  002374  052350                        DT601
```

```
2029  002376  053144                          DF21
2030
2031                                          ;ERROR 100
2032  002400  047710                          EM77                    ;RETRY SUCCESFUL MESSAGE
2033  002402  000000                          0
2034  002404  052350                          DT601
2035  002406  053204                          DF23
2036
2037                                          ;ERROR 101
2038  002410  047710                          EM77                    ;ANOTHER RETRY SUCCESSFUL MESSAGE
2039  002412  000000                          0
2040  002414  052350                          DT601
2041  002416  053204                          DF23
2042
2043                                          ;ERROR 102
2044  002420  047731                          EM100                   ;RETRY UNSUCCESSFUL MESSAGE
2045  002422  000000                          0
2046  002424  052350                          DT601
2047  002426  053204                          DF23
2048
2049                                          ;ERROR 103
2050  002430  047754                          EM101                   ;NO VALID HEADERS IN TRACK JUST READ
2051  002432  051447                          DH6042
2052  002434  052350                          DT601
2053  002436  053214                          DF24
2054
2055                                          ;ERROR 104
2056  002440  050032                          EM102                   ;BSE ERROR ON SECTOR NOT LISTED AS BAD
2057  002442  050500                          DH100
2058  002444  052264                          DT100
2059  002446  052424                          DF02
2060
2061                                          ;ERROR  105
2062  002450  050104                          EM103                   ;TIMED-OUT ON READ HEADER
2063  002452  050500                          DH100
2064  002454  052264                          DT100
2065  002456  052460                          DF03
2066
2067                                          ;ERROR  106
2068  002460  050132                          EM104                   ;TIMED-OUT ON SEEK
2069  002462  050500                          DH100
2070  002464  052264                          DT100
2071  002466  052460                          DF03
2072
2073                                          ;ERROR  107
2074  002470  050154                          EM105                   ;DRIVE SIEZED BY OTHER PORT
2075  002472  050500                          DH100
2076  002474  052264                          DT100
2077  002476  052424                          DF02
2078
2079                                          ;ERROR  110
2080  002500  050207                          EM106                   ;''DATA MISCMPR WHILE BAI SET''
2081  002502  050500                          DH100
2082  002504  052264                          DT100
2083  002506  053234                          DF27
2084
```

```
2085                                    ;ERROR  111
2086  002510  050242                    EM107                   ;'NO NEM WHEN EXPECTED''
2087  002512  000000                    0
2088  002514  000000                    0
2089  002516  000000                    0
2090
2091                                    ;ERROR  112
2092  002520  050307                    EM110                   ;''INTRPT WHEN CNTRLR NOT READY''
2093  002522  050500                    DH100
2094  002524  052264                    DT100
2095  002526  052374                    DF01
2096
2097                                    ;ERROR  113
2098  002530  050342                    EM111                   ;'NO ATT'N ON SEEK''
2099  002532  050500                    DH100
2100  002534  052264                    DT100
2101  002536  052424                    DF02
2102
2103                                    ;ERROR  114
2104  002540  050363                    EM112                   ;DRIVE'S CYLINDER INCORRECT
2105  002542  051674                    DH702
2106  002544  052324                    DT202
2107  002546  053224                    DF26
2108
2109                                    ;ERROR  115
2110  002550  000000                    0                       ;TYPE ADRS OF DATA MiSCOMPARE(S)
2111  002552  000000                    0
2112  002554  052350                    DT601
2113  002556  052650                    DF11
2114
2115                                    ;ERROR  116
2116  002560  046333                    EM34                    ;DATA MISCOMPARE (11/70)
2117  002562  051011                    DH103
2118  002564  000000                    0
2119  002566  053134                    DF20
2120
2121                                    ;ERROR  117
2122  002570  000000                    0                       ;PART OF DATA MISCOMPARE
2123  002572  000000                    0
2124  002574  052264                    DT100
2125  002576  053154                    DF22
2126
2127                                    ;ERROR  120
2128  002600  050416                    EM113                   ;ABORT- CAN'T READ BSF
2129  002602  050500                    DH100
2130  002604  052264                    DT100
2131  002606  052374                    DF01
2132
2133                                    ;ERROR  121
2134  002610  050444                    EM114                   ;KT11 FAILURE
2135  002612  050500                    DH100
2136  002614  052264                    DT100
2137  002616  053260                    DF30
2138
2139                                    ;ERROR  122
2140  002620  050461                    EM115                   ;MEM PARITY ERROR
```

```
  2141  002622  050500                        DH100
  2142  002624  052264                        DT100
  2143  002626  053304                        DF31
  2144
  2145                                        ;ERROR  123
  2146  002630  045455                        EM3                     ;NED ON SIZING UNDER APT
  2147  002632  000000                        0
  2148  002634  000000                        0
  2149  002636  000000                        0
  2150
  2151
  2152
  2153
  2154
  2155
  2156
  2157
  2158
  2159
  2160
  2161
  2162
  2163
  2164
  2165
  2166
  2167
  2168
  2169
```

```
                        .SBTTL  BIT ASSIGNMENTS IN THE RK611 REGISTERS

                        ;RKCS1
                        ; 15     14     13     12     11     10     9      8
                        ;-------------------------------------------------------
                        ;CERR ! DI   ! DCPAR! CFMT ! CTO  ! CDT  ! BA17 ! BA16 !
                        ;CCLR !      !      !      !      !      !      !      !
                        ; R/W ! RO   ! RO   ! R/W  ! RO   ! R/W  ! R/W  ! R/W  !
                        ;-------------------------------------------------------

                        ;  7     6      5      4      3      2      1      0
                        ;-------------------------------------------------------
                        ; RDY ! IE   ! SPARE! F4   ! F3   ! F2   ! F1   ! GO   !
                        ; R/W ! R/W  !  R/W ! R/W  ! R/W  ! R/W  ! R/W  ! R/W  !
                        ;-------------------------------------------------------

                        ;RKCS2
                        ; 15    14     13     12     11     10     9      8
                        ;-------------------------------------------------------
                        ; DLT ! WCE  ! UPE  ! NED  ! NEM  ! PGE  ! MDS  ! UFE  !
                        ; RO  ! RO   ! RO   ! RO   ! RO   ! RO   ! RO   ! RO   !
                        ;-------------------------------------------------------

                        ;  7     6      5      4      3      2      1      0
                        ;-------------------------------------------------------
                        ; OR  ! IR   ! CLR  ! BAI  ! DESL ! DS2  ! DS1  ! DS0  !
                        ; RO  ! RO   ! WO   ! R/W  ! R/W  ! R/W  ! R/W  ! R/W  !
                        ;-------------------------------------------------------

                        ;RKDC (DESIRED CYLINDER)
                        ; 15     14     13     12     11     10     9      8
                        ;-------------------------------------------------------
                        ; NU  !  NU  !  NU  !  NU  !  NU  !  NU  ! DC9  ! DC8  !
                        ;                    READ ONLY                          !
                        ;-------------------------------------------------------

                        ;  7      6      5      4      3      2      1      0
                        ;-------------------------------------------------------
                        ; DC7 ! DC6  ! DC5  ! DC4  ! DC3  ! DC2  ! DC1  ! DC0  !
                        ;                    READ ONLY                          !
                        ;-------------------------------------------------------

                        ;RKDA (DESIRED TRACK AND SECTOR)
                        ; 15     14     13     12     11     10     9      8
                        ;-------------------------------------------------------
                        ; NU  !  UN  !  UN  !  UN  !  UN  ! TA2  ! TA1  ! TA0  !
                        ;                    READ ONLY                          !
                        ;-------------------------------------------------------

                        ;  7      6      5      4      3      2      1      0
                        ;-------------------------------------------------------
                        ; UN  !  UN  !  UN  ! SA4  ! SA3  ! SA2  ! SA1  ! SA0  !
                        ;                    READ ONLY                          !
                        ;-------------------------------------------------------
```

```
2224                                    ;RKDB (DATA BUFFER)
2225                                    ; 15     14     13     12     11     10     9      8
2226                                    ;-----------------------------------------------------
2227                                    ;DB15 ! DB14 ! DB13 ! DB12 ! DB11 ! DB10 ! DB9  ! DB8  !
2228                                    ;                    READ/WRITE                       !
2229                                    ;-----------------------------------------------------
2230
2231                                    ;  7      6      5      4      3      2      1      0
2232                                    ;-----------------------------------------------------
2233                                    ; DB7 ! DB6  ! DB5  ! DB4  ! DB3  ! DB2  ! DB1  ! DB0  !
2234                                    ;                    READ/WRITE                       !
2235                                    ;-----------------------------------------------------
2236
2237                                    ;RKASOF (ATTENTION SUMMARY AND OFFSET)
2238                                    ; 15     14     13     12     11     10     9      8
2239                                    ;-----------------------------------------------------
2240                                    ;ATN7 ! ATN6 ! ATN5 ! ATN4 ! ATN3 ! ATN2 ! ATN1 ! ATN0 !
2241                                    ;                    READ ONLY                        !
2242                                    ;-----------------------------------------------------
2243
2244                                    ;  7      6      5      4      3      2      1      0
2245                                    ;-----------------------------------------------------
2246                                    ; UN  ! OF6  : OF5  ! OF4  ! OF3  ! OF2  ! OF1  ! OF0  !
2247                                    ;  ?  !              READ/WRITE                       !
2248                                    ;-----------------------------------------------------
2249
2250                                    ;RKWC (WORD COUNT)
2251                                    ; 15     14     13     12     11     10     9      8
2252                                    ;-----------------------------------------------------
2253                                    ;WC15 ! WC14 ! WC13 ! WC12 ! WC11 ! WC10 ! WC9  ! WC8  !
2254                                    ;                    READ/WRITE                       !
2255                                    ;-----------------------------------------------------
2256
2257                                    ;  7      6      5      4      3      2      1      0
2258                                    ;-----------------------------------------------------
2259                                    ; WC7 ! WC6  ! WC5  ! WC4  ! WC3  ! WC2  ! WC1  ! WC0  !
2260                                    ;                    READ/WRITE                       !
2261                                    ;-----------------------------------------------------
2262
2263                                    ;RKBA (BUS ADDRESS)
2264                                    ; 15     14     13     12     11     10     9      8
2265                                    ;-----------------------------------------------------
2266                                    ;BA15 ! BA14 ! BA13 ! BA12 ! BA11 ! BA10 ! BA9  ! BA8  !
2267                                    ;                    READ/WRITE                       !
2268                                    ;-----------------------------------------------------
2269
2270                                    ;  7      6      5      4      3      2      1      0
2271                                    ;-----------------------------------------------------
2272                                    ; BA7 ! BA6  ! BA5  ! BA4  ! BA3  ! BA2  ! BA1  ! BA0  !
2273                                    ;                    READ/WRITE                !ALWYS0!
2274                                    ;-----------------------------------------------------
```

```
                        ;RKER (ERROR REGISTER)
                        ; 15     14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ; DCK ! UNS  ! OPI  ! DTE  ! WLE  ! IDAE ! COE  ! HVRC !
                        ;                    READ ONLY                        !
                        ;-----------------------------------------------------

                        ;  7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ; BSE ! ECH  ! DTYPE! FMTE ! DRPAR! ILF  ! SKI  ! ILC  !
                        ;                    READ ONLY                        !
                        ;-----------------------------------------------------

                        ;RKDS (DRIVE STATUS)
                        ; 15     14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ;SVAL ! DSC  ! PIP  ! SPON ! WRL  !  UN  !  UN  ! DTP  !
                        ;                    READ ONLY                        !
                        ;-----------------------------------------------------

                        ;  7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ;DRDY !  VV  ! DROT ! SPLS ! ACLO ! OFSET!  UN  ! DRA  !
                        ;                    READ ONLY                        !
                        ;-----------------------------------------------------

                        ;RKMR1 (MAINTENANCE REG 1)
                        ; 15    14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ;RD   ! WRT  ! ECCW ! PCD  ! PCA  ! MEWD ! MERD ! MCLK !
                        ;GATE ! GATE !      !      !      !      !      !      !
                        ;                READ ONLY                ! READ/WRITE  !
                        ;-----------------------------------------------------

                        ;  7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ;MIND ! MSP  ! DMD  ! PAT  ! MS3  ! MS2  ! MS1  ! MS0  !
                        ;                  READ/WRITE                         !
                        ;-----------------------------------------------------
```

```
2314                                    ;RKECC/PAT
2315                                    ; 15     14     13     12     11     10     9      8
2316                                    ;-----------------------------------------------------
2317                                    ; UN  !  UN  !  UN  !  UN  !  UN  ! EPA10! EPA9 ! EPA8 !
2318                                    ;                    READ ONLY                        !
2319                                    ;-----------------------------------------------------
2320
2321                                    ;  7      6      5      4      3      2      1      0
2322                                    ;-----------------------------------------------------
2323                                    ;EPA7 ! EPA6 ! EPA5 ! EPA4 ! EPA3 ! EPA2 ! EPA1 ! EPA0 !
2324                                    ;                    READ ONLY                        !
2325                                    ;-----------------------------------------------------
2326
2327                                    ;RKECC/POS
2328                                    ; 15     14     13     12     11     10     9      8
2329                                    ;-----------------------------------------------------
2330                                    ; UN  !  UN  !  UN  ! EPO12! EPO11! EPO10! EPO9 ! EPO8 !
2331                                    ;                    READ ONLY                        !
2332                                    ;-----------------------------------------------------
2333
2334                                    ;  7      6      5      4      3      2      1      0
2335                                    ;-----------------------------------------------------
2336                                    ;EPO7 ! EPO6 ! EPO5 ! EPO4 ! EPO3 ! EPO2 ! EPO1 ! EPO0 !
2337                                    ;                    READ ONLY                        !
2338                                    ;-----------------------------------------------------
2339
2340                                    ;DRIVE STATUS INFORMATION
2341
2342                                    ;LINE A MESSAGE 00
2343                                    ; 15     14     13     12     11     10     9      8
2344                                    ;-----------------------------------------------------
2345                                    ; PAR ! DSC  ! PIP  ! SPON ! WRLK ! OFFON! FMT  ! DRTP !
2346                                    ;-----------------------------------------------------
2347
2348                                    ;  7      6      5      4      3      2      1      0
2349                                    ;-----------------------------------------------------
2350                                    ;DRDY !  VV  ! DRA  !  NU  !  NU  ! DRIVE SELECT CODE  !
2351                                    ;-----------------------------------------------------
```

```
                        ;LINE B MESSAGE 00
                        ; 15     14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ; PAR ! UNS  ! DROT ! DCLO ! WLE  ! SKI  ! PERR ! ILF  !
                        ;-----------------------------------------------------
                        ;
                        ; 7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ; FLT ! ACLO ! IVDA !  UN  !  UN  !  UN  !  0   !  0   !
                        ;-----------------------------------------------------

                        ;LINE A MESSAGE 01
                        ; 15     14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ; PAR ! UNLDG! RTZ  ! LDG  ! REV  ! FWD  ! SPEED! CART !
                        ;     ! HDS  !      ! HDS  !      !      ! OK   ! PRES !
                        ;-----------------------------------------------------

                        ; 7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ;DOOR ! BRUSH! HEADS! TRK  !  UN  !  DRIVE SELECT CODE !
                        ;LTCH ! HOME ! HOME ! FOLOW!      !                     !
                        ;     !      !      !  OK  !      !                     !
                        ;-----------------------------------------------------

                        ;LINE B MESSAGE 01
                        ; 15     14     13     12     11     10     9      8
                        ;-----------------------------------------------------
                        ; PAR ! SERVO! LIMIT! SEEK ! PLO  ! DIBIT! INDEX! MULTI!
                        ;     ! BRAKE! ON SK!NO MO ! ERR  !  ERR ! ERR  !HD SEL!
                        ;-----------------------------------------------------

                        ; 7      6      5      4      3      2      1      0
                        ;-----------------------------------------------------
                        ;HEAD !WR GTE!WR CUR! SEC  !  NU  !  NU  !  0   !  1   !
                        ;FAULT!AND NO!AND NO! ERR  !      !      !      !      !
                        ;     !XISTON!WR GTE!      !      !      !      !      !
                        ;-----------------------------------------------------
```

```
                                        ;LINE A MESSAGE 10
                                        ; 15     14     13     12     11     10     9      8
                                        ;-----------------------------------------------------
                                        ; PAR !  NU  !  NU  ! CYLINDER DIFF/OFFSET VALUE      !
                                        ;-----------------------------------------------------

                                        ;  7      6      5      4      3      2      1      0
                                        ;-----------------------------------------------------
                                        ;CYLINDER DIFF/OFFSET VALUE!  NU  !  DRIVE SELECT CODE !
                                        ;-----------------------------------------------------

                                        ;LINE B MESSAGE 10
                                        ; 15     14     13     12     11     10     9      8
                                        ;-----------------------------------------------------
                                        ; PAR ! ALIGN!  NU  !    CYLINDER ADDRESS             !
                                        ;     ! SIGN !                                        !
                                        ;-----------------------------------------------------

                                        ;  7      6      5      4      3      2      1      0
                                        ;-----------------------------------------------------
                                        ;    CYLINDER ADDRESS      ! UN  ! UN  !   1  !   0  !
                                        ;-----------------------------------------------------

                                        ;LINE A MESSAGE 11
                                        ; 15     14     13     12     11     10     9      8
                                        ;-----------------------------------------------------
                                        ; PAR !     DRIVE SERIAL NUMBER                       !
                                        ;-----------------------------------------------------

                                          7      6      5      4      3      2      1      0
                                        ;-----------------------------------------------------
                                        ; DRIVE SERIAL NUMBER             !  DRIVE SELECT CODE !
                                        ;-----------------------------------------------------

                                        ;LINE B MESSAGE 11
                                        ; 15     14     13     12     11     10     9      8
                                        ;-----------------------------------------------------
                                        ; PAR !  NU  !    NU       !   DECODED HEAD      ! SECTOR!
                                        ;     !      !             !    ADDRESS          ! COUNT !
                                        ;-----------------------------------------------------

                                        ;  7      6      5      4      3      2      1      0
                                        ;-----------------------------------------------------
                                        ;   SECTOR COUNT           ! UN  ! UN  !   1  !   1  !
                                        ;-----------------------------------------------------


                                .SBTTL  RK06 CONTROLLER REGISTER DEFINITION

000000                          RKCS1=  0               ;CONTROL AND STATUS REGISTER 1
000002                          RKWC=   2               ;WORD COUNT REGISTER
000004                          RKBA=   4               ;BUS ADDRESS REGISTER
000006                          RKDA=   6               ;DESIRED TRACK SECTOR REGISTER
000010                          RKCS2=  10              ;CONTROL AND STATUS REGISTER 2
```

```
                                ;DRIVE STATUS REGISTER
                                ;ERROR REGISTER
                                ;ATTENTION SUMMARY AND OFFSET REGISTER
                                ;DESIRED CYLINDER REGISTER
                                ;DESIRED CYLINDER REGISTER
                                ;DATA BUFFER
                                ;MAINTENANCE REGISTER 1
                                ;MAINTENANCE REGISTER 2
                                ;MAINTENANCE REGISTER 3
                                ;ECC POSITION INFORMATION
                                ;ECC POSITION INFORMATION
                                ;ECC PATTERN INFORMATION
                                ;ECC PATTERN INFORMATION

                        .SBTTL  DRIVE COMMANDS

                                ;SELECT DRIVE
                                ;PACK ACKNOWLEDGE
                                ;DRIVE CLEAR
                                ;UNLOAD
                                ;START SPINDLE
                                ;RECALIBRATE
                                ;OFFSET
                                ;SEEK
                                ;READ DATA
                                ;WRITE DATA
                                ;READ HEADER
                                ;WRITE HEADER AND DATA
                                ;WRITE CHECK

                        ;       THE FOLLOWING ARE NOT DRIVE COMMANDS BUT ARE USED BY THE DRIVER
                        ;       TO SIMULATE A SPECIFIC DESIRED OPERATION

                                ;RELEASE DRIVE
                                ;GET ALL STATUS FROM DRIVE
                                ;READ ALL HEADERS
                                ;CONTROLLER CLEAR (BIT 15 OF CS1)
                                ;SUBSYSTEM CLEAR (BIT 5 OF CS2)
                                ;GENERATE INTERRUPT TO CPU

                        ;       DRIVER ISSUED SERVICE COMMANDS

                                ;DRIVE SELECT
                                ;DRIVE CLEAR

                        .SBTTL  CONTROL AND STATUS REGISTER 1 BITS

                                ;GO BIT
                                ;INTERRUPT ENABLE
                                ;CONTROLLER READY
                                ;BUS ADDRESS BIT 16
                                ;BUS ADDRESS BIT 17
                                ;CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
                                ;CONTROLLER TIMED OUT WAITING FOR
                                ;  DRIVE RESPONSE
                                ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
```

CONTROL AND STATUS REGISTER 1 BITS SEQ 0052

```
                        020000          SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
                        040000          DI=     BIT14                   ;DRIVE INTERRUPT
                        100000          CERR=   BIT15                   ;CONTROLLER ERROR
                        100000          CCLR=   BIT15                   ;CONTROLLER CLEAR

                                        ;       THESE BIT DEFINITIONS APE USED FOR ADDRESS
                                        ;       THE HIGH BYTE OF RKCS1

                        000001          B.BA16= BIT0                    ;BUS ADDRESS BIT 16
                        000002          B.BA17= BIT1                    ;BUS ADDRESS BIT 17
                        000004          B.CDT=  BIT2                    ;CONTROLLER DRIVE TYPE (0=RK06,1=RK07)
                        000020          B.CFMT= BIT4                    ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)

                                        .SBTTL  CONTROL AND STATUS REGISTER 2 BITS

                        000007          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
                        000010          DESL=   BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
                        000010          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
                        000020          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
                        000040          CLR=    BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
                        000040          SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
                        000100          IR=     BIT6                    ;INPUT READY
                        000200          OR=     BIT7                    ;OUTPUT READY
                        000400          UFE=    BIT8                    ;UNIT FIELD ERROR
                        001000          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
                        002000          PGE=    BIT10                   ;PROGRAMMING ERROR
                        004000          NEM=    BIT11                   ;NON-EXISTENT MEMORY
                        010000          NED=    BIT12                   ;NON-EXISTENT DRIVE
                        020000          UPE=    BIT13                   ;UNIBUS PARITY ERROR
                        040000          WCE=    BIT14                   ;WRITE CHECK ERROR
                        100000          DLT=    BIT15                   ;DATA LATE ERROR

                                        .SBTTL  ERROR REGISTER BIT DEFINITION

                        000001          ILC=    BIT0                    ;ILLEGAL FUNCTION CODE
                                        ;*ILF=  BIT0                    ;ILLEGAL FUNCTION CODE
                        000002          SKI=    BIT1                    ;SEEK INCOMPLETE
                        000004          ILF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
                        000004          NXF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
                        000010          DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
                        000020          FMTE=   BIT4                    ;FORMAT ERROR
                        000040          DTYE=   BIT5                    ;DRIVE TYPE ERROR
                        000100          ECH=    BIT6                    ;ECC HARD
                        000200          BSE=    BIT7                    ;BAD SECTOR ERROR
                        000400          HCRC=   BIT8                    ;HEADER CRC ERRROR
                        000400          HVRC=   BIT8                    ;HEADER VRC ERRROR
                        001000          COE=    BIT9                    ;CYLINDER ADDRESS OVERFLOW ERROR
                        002000          IDAE=   BIT10                   ;INVALID DISK ADDRESS ERROR
                        004000          WLE=    BIT11                   ;WRITE LOCK ERROR
                        010000          DTE=    BIT12                   ;DRIVE TIMING ERROR
                        020000          OPI=    BIT13                   ;OPERATION (SEARCH) INCOMPLETE
                        040000          UNS=    BIT14                   ;DRIVE UNSAFE
                        100000          DCK=    BIT15                   ;DATA CHECK

                                        .SBTTL  STATUS REGISTER BIT DEFINITION
```

```
                        CRA=    BIT0                    ;DRIVE AVAILABLE (CONTROLLER IS SET IF
                                                        ;  THIS BIT IS RESET)
000004                  OFST=   BIT2                    ;DRIVE OFFSET
000010                  ACLO=   BIT3                    ;AC LOW
000020                  SPDLSS= BIT4                    ;SPEED LOSS
000020                  DCLO=   BIT4                    ;DC LOW
000040                  DROT=   BIT5                    ;DRIVE OFF TRACK
000100                  VV=     BIT6                    ;VOLUME VALID
000200                  DRY=    BIT7                    ;DRIVE READY
000200                  DRDY=   BIT7                    ;DRIVE READY
000400                  DDT=    BIT8                    ;DRIVE TYPE (0=RK06,1=RK07)
004000                  WRL=    BIT11                   ;WRITE LOCK
020000                  PIP=    BIT13                   ;POSITIONING IN PROGRESS
040000                  DSC=    BIT14                   ;DRIVE STATUS CHANGE
100000                  SVAL=   BIT15                   ;STATUS VALID

                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION

000017                  MESMSK= 17                      ;MESSAGE MASK

000020                  PAT=    BIT4                    ;FORCE EVEN PARITY ON DRIVE BUS MESSAGE LINES
000040                  DMD=    BIT5                    ;DIAGNOSTIC MODE
000100                  MSP=    BIT6                    ;MAINTENANCE SECTOR PULSE
000200                  MIND=   BIT7                    ;MAINTENANCE INDEX
000400                  MCLK=   BIT8                    ;MAINTENANCE CLOCK
001000                  MERD=   BIT9                    ;MAINTENANCE ENCODED READ DATA
002000                  MEWD=   BIT10                   ;MAINTENACNE ENCODED WRITE DATA
004000                  PCA=    BIT11                   ;PRECOMPENSATION ADVANCE
010000                  PCD=    BIT12                   ;PRECOMPENSATION DELAY
020000                  ECCW=   BIT13                   ;ECC WORD IS BEING READ OR WRITTEN
040000                  WRTGAT= BIT14                   ;WRITE GATE
100000                  RDGATE= BIT15                   ;READ GATE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A

000040                  S.DRA=  BIT5                    ;DRIVE AVAILIABLE
000100                  S.VV=   BIT6                    ;VOLUME VALID
000200                  S.DRY=  BIT7                    ;DRIVE READY
000400                  S.TYPE= BIT8                    ;DRIVE TYPE
001000                  S.FORM= BIT9                    ;DRIVE FORMAT
002000                  S.OFF=  BIT10                   ;OFFSET
004000                  S.WRL=  BIT11                   ;WRITE LOCK
010000                  S.SPIN= BIT12                   ;SPINDLE ON
020000                  S.PIP=  BIT13                   ;POSITIONING IN PROGRESS
040000                  S.DSC=  BIT14                   ;DRIVE STATUS CHANGE

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B

000040                  S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
000100                  S.ACLO= BIT6                    ;AC LOW
000200                  S.FLT=  BIT7                    ;DRIVE FAULT
000400                  S.ILF=  BIT8                    ;ILLEGAL FUNCTION
001000                  S.PAR=  BIT9                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
002000                  S.SKI=  BIT10                   ;SEEK INCOMPLETE
004000                  S.WLE=  BIT11                   ;WRITE LOCK ERROR
010000                  S.SPLS= BIT12                   ;SPEED LOSS
```

```
                        DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE B                                    SEQ 0054

2614            010000                  S.DCLO= BIT12                   ;DC LOW
2615            020000                  S.DROT= BIT13                   ;DRIVE OFF TRACK
2616            040000                  S.UNS=  BIT14                   ;DRIVE UNSAFE
2617
2618                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
2619
2620            000020                  S.XDOK= BIT4                    ;TRANSDUCER OK
2621            000040                  S.HDHM= BIT5                    ;HEADS HOME
2622            000100                  S.BRHM= BIT6                    ;BRUSHES HOME
2623            000200                  S.DOOR= BIT7                    ;DOOR INTERLOCKED
2624            000400                  S.CART= BIT8                    ;CARTRAGE INTERLOCK
2625            001000                  S.SPOK= BIT9                    ;SPEED OK
2626            002000                  S.FWD=  BIT10                   ;FORWARD
2627            004000                  S.REV=  BIT11                   ;REVERSE
2628            010000                  S.LOAD= BIT12                   ;HEADS LOADING
2629            020000                  S.RTZ=  BIT13                   ;RETURN TO ZERO
2630            040000                  S.UNLD= BIT14                   ;HEADS UNLOADING
2631
2632                                    .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
2633
2634            000020                  S.SECT= BIT4                    ;SECTOR ERROR
2635            000040                  S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
2636            000100                  S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
2637            000200                  S.HDFL= BIT7                    ;HEAD FAULT
2638            000400                  S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
2639            001000                  S.XERR= BIT9                    ;INDEX ERROR
2640            002000                  S.DIB=  BIT10                   ;DIBIT ERROR
2641            004000                  S.PLO=  BIT11                   ;PLO ERROR
2642            010000                  S.NMOV= BIT12                   ;SEEK AND NO MOTION
2643            020000                  S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
2644            040000                  S.BRKE= BIT14                   ;SERVO-BRAKE
2645
2646                                    .SBTTL  COMMON MASKS
2647
2648            000007                  M.DRV=  7                       ;DRIVE CODE
2649            100000                  M.PAR=  BIT15                   ;PARITY
2650            000003                  M.ID=   3                       ;BYTE ID
2651            017760                  M.CDIF= 17760                   ;CYLINDER DIFFERENCE/OFFSET
2652            017760                  M.CADD= 17760                   ;CYLINDER ADDRESS
2653            077770                  M.SER=  77770                   ;DRIVE SERIAL NUMBER
2654            000760                  M.SECT= 760                     ;SECTOR COUNT
2655            007000                  M.HEAD= 7000                    ;HEAD DECODE
```

```
                                .SBTTL  PARAMETER BLOCK ALLOCATION

                        ;*      ************************************************
                        ;*  1   ! COMMAND                 ! DRIVE NO.          !  0
                        ;*  3   ! CYLINDER ADDRESS                             !  2
                        ;*  5   ! TRACK                   ! SECTOR             !  4
                        ;*  7   !BA16-17,FORMAT,DRV TYPE! OFFSET               !  6
                        ;* 11   ! BUS ADDRESS (LOW 16 BITS)                    ! 10
                        ;* 13   ! WORD COUNT (2'S COMPLEMENT)                  ! 12
                        ;* 15   ! PROGRAM DRIVE STATUS INFORMATION             ! 14
                        ;* 17   ! COMMAND AND STATUS REGISTER 1                ! 16
                        ;* 21   ! COMMAND AND STATUS REGISTER 2                ! 20
                        ;* 23   ! WORD COUNT REGISTER                          ! 22
                        ;* 25   ! BUS ADDRESS REGISTER                         ! 24
                        ;* 27   ! DESIRED TRACK AND SECTOR                     ! 26
                        ;* 31   ! DESIRED CYLINDER                             ! 30
                        ;* 33   ! ATTENTION SUMMARY AND DRIVE OFFSET           ! 32
                        ;* 35   ! ERROR REGISTER                               ! 34
                        ;* 37   ! STATUS REGISTER                              ! 36
                        ;* 41   ! MESSAGE LINE A STATUS BYTE 00                ! 40
                        ;* 43   ! MESSAGE LINE B STATUS BYTE 00                ! 42
                        ;* 45   ! MESSAGE LINE A STATUS BYTE 01                ! 44
                        ;* 47   ! MESSAGE LINE B STATUS BYTE 01                ! 46
                        ;* 51   ! MESSAGE LINE A STATUS BYTE 10                ! 50
                        ;* 53   ! MESSAGE LINE B STATUS BYTE 10                ! 52
                        ;* 55   ! MESSAGE LINE A STATUS BYTE 11                ! 54
                        ;* 57   ! MESSAGE LINE B STATUS BYTE 11                ! 56
                        ;* 61   ! ECC POSITION INFORMATION                     ! 60
                        ;* 63   ! ECC PATTERN INFORMATION                      ! 62
                        ;*      ************************************************

                                .SBTTL  PARAMETERS PASSED TO THE DRIVER

                                ;       THE FOLLOWING DEFINITIONS ARE USED TO PASS PARAMETERS
                                ;         TO THE RK06/RK07 DRIVER

000000                          P.DRVN= 0                       ;DRIVE NUMBER
000001                          P.CMND= 1                       ;COMMAND
000002                          P.CYLN= 2                       ;CYLINDER ADDRESS
000004                          P.SECT= 4                       ;SECTOR
000005                          P.TRCK= 5                       ;TRACK
000006                          P.OFST= 6                       ;OFFSET
000007                          P.CS1H= 7                       ;RKCS1 BITS 8-15
000007                          P.BAHI= 7                       ;BUS ADDRESS (BITS 16 AND 17)
000010                          P.BALO= 10                      ;BUS ADDRESS (BITS 0-15)
000012                          P.WC=   12                      ;WORD COUNT (2'S COMPLEMENT)
000014                          P.PRST= 14                      ;PROGRAM DRIVE STATUS INFORMATION

                                .SBTTL  PROGRAM DEVICE STATUS REGISTER DEFINITION

000001                          DRVUSE= BIT0                    ;DRIVE IN USE
000002                          DRVPOS= BIT1                    ;DRIVE POSITIONING
000004                          DRVPDT= BIT2                    ;DRIVE POSITIONED FOR DATA TRANSFER
000010                          UEXATT= BIT3                    ;UNEXPECTED ATTENTION
000020                          DRVHRD= BIT4                    ;DRIVE HAS HARD ERROR
000040                          DRVDSC= BIT5                    ;DRIVE STATUS CHANGE DID NOT CLEAR
```

```
                        PROGRAM DEVICE STATUS REGISTER DEFINITION                                   SEQ 0056

2712            000100                  CMDTO=  BIT6                    ;NO TERMINATION TO COMMAND FOR AT
2713                                                                    ;  LEAST 1 SECOND
2714            000200                  W.WCK=  BIT7                    ;WRITE FOR WRITE WRITE CHECK
2715            000400                  NOCHK=  BIT8                    ;NO CHECK, DO NOT SET INTERRUPT ENABLE
2716            001000                  PBSVAL= BIT9                    ;PARAMETER STATUS WORDS VALID
27.7                                                                    ;  (SET WHEN ERROR TERMINATION OR
2718                                                                    ;  READ STATUS COMMAND)
2719            002000                  DRPDRV= BIT10                   ;DROP DRIVE FROM TEST SEQUENCE
2720            004000                  NODSC=  BIT11                   ;ATTENTION SET BUT DCS AND FAULT RESET
2721            010000                  DRVSZD= BIT12                   ;DRIVE SEIZED BY OTHER PORT
2722            020000                  E.UNLD= BIT13                   ;DRIVE UNLOADED DUE TO ERROR
2723            040000                  Q.INIT= BIT14                   ;PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
2724            100000                  DTBAII= BIT15                   ;INHIBIT BUS ADDRESS INCREMENT
2725
2726                                    .SBTTL  PARAMETERS PASSED FROM DRIVER TO PROGRAM
2727
2728                                    ;       THE FOLLOWING DEFINITIONS ARE USED FOR REGISTER RETURNS
2729                                    ;         FROM THE DRIVER TO THE CALLING PROGRAM
2730
2731            000016                  P.CS1=  16                      ;COMMAND AND STATUS REGISTER 1
2732            000020                  P.CS2=  20                      ;COMMAND AND STATUS REGISTER 2
2733            000022                  P.WCR=  22                      ;WORD COUNT REGISTER
2734            000024                  P.BAR=  24                      ;BUS ADDRESS REGISTER
2735            000026                  P.DTS=  26                      ;DESIRED TRACK SECTOR REGISTER
2736            000030                  P.DCYL= 30                      ;DESIRED CYLINDER REGISTER
2737            000032                  P.ASOF= 32                      ;ATTENTION SUMMARY/OFFSET REGISTER
2738            000034                  P.ER=   34                      ;ERROR REGISTER
2739            000036                  P.DS=   36                      ;STATUS REGISTER
2740            000040                  P.A00=  40                      ;MESSAGE A STATUS BYTE 00
2741            000042                  P.B00=  42                      ;MESSAGE B STATUS BYTE 00
2742            000044                  P.A01=  44                      ;MESSAGE A STATUS BYTE 01
2743            000046                  P.B01=  46                      ;MESSAGE B STATUS BYTE 01
2744            000050                  P.A10=  50                      ;MESSGGE A STATUS BYTE 10
2745            000052                  P.B10=  52                      ;MESSAGE B STATUS BYTE 10
2746            000054                  P.A11=  54                      ;MESSAGE A STATUS BYTE 11
2747            000056                  P.B11=  56                      ;MESSAGE B STATUS BYTE 11
2748            000060                  P.EPOS= 60                      ;ECC POSITION INFORMATION
2749            000062                  P.EPAT= 62                      ;ECC PATTERN INFORMATION
2750
2751                                    .SBTTL  PARAMETER BLOCK 0 FOR DRIVE
2752
2753    002640      000                 PARM0:  .BYTE   0               ;DRIVE NUMBER
2754    002641      000                         .BYTE   0               ;COMMAND
2755    002642   000000                         .WORD   0               ;CYLINDER ADDRESS
2756    002644      000                         .BYTE   0               ;SECTOR ADDRESS
2757    002645      000                         .BYTE   0               ;TRACK ADDRESS
2758    002646      000                         .BYTE   0               ;OFFSET VALUE
2759    002647      000                         .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
2760    002650   000000                         .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
2761    002652   000000                         .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
2762    002654   000000                         .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
2763    002656   000000                         .WORD   0               ;COMMAND AND STATUS REGISTER 1
2764    002660   000000                         .WORD   0               ;COMMAND AND STATUS REGISTER 2
2765    002662   000000                         .WORD   0               ;WORD COUNT REGISTER
2766    002664   000000                         .WORD   0               ;BUS ADDRESS REGISTER
2767    002666   000000                         .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
```

```
 2768  002670  000000                        .WORD   0               ;DESIRED CYLINDER REGISTER
 2769  002672  000000                        .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
 2770  002674  000000                        .WORD   0               ;ERROR REGISTER
 2771  002676  000000                        .WORD   0               ;STATUS REGISTER
 2772  002700  000000                        .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
 2773  002702  000000                        .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
 2774  002704  000000                        .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
 2775  002706  000000                        .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
 2776  002710  000000                        .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
 2777  002712  000000                        .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
 2778  002714  000000                        .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
 2779  002716  000000                        .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
 2780  002720  000000                        .WORD   0               ;ECC POSITION INFORMATION
 2781  002722  000000                        .WORD   0               ;ECC PATTERN INFORMATION
 2782
 2783                                .SBTTL  PARAMETER BLOCK 1 FOR DRIVE
 2784
 2785  002724     000              PARM1:  .BYTE   0               ;DRIVE NUMBER
 2786  002725     000                      .BYTE   0               ;COMMAND
 2787  002726  000000                      .WORD   0               ;CYLINDER ADDRESS
 2788  002730     000                      .BYTE   0               ;SECTOR ADDRESS
 2789  002731     000                       BYTE   0               ;TRACK ADDRESS
 2790  002732     000                      .BYTE   0               ;OFFSET VALUE
 2791  002733     000                      .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
 2792  002734  000000                      .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
 2793  002736  000000                      .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
 2794  002740  000000                      .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
 2795  002742  000000                      .WORD   0               ;COMMAND AND STATUS REGISTER 1
 2796  002744  000000                      .WORD   0               ;COMMAND AND STATUS REGISTER 2
 2797  002746  000000                      .WORD   0               ;WORD COUNT REGISTER
 2798  002750  000000                      .WORD   0               ;BUS ADDRESS REGISTER
 2799  002752  000000                      .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
 2800  002754  000000                      .WORD   0               ;DESIRED CYLINDER REGISTER
 2801  002756  000000                      .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
 2802  002760  000000                      .WORD   0               ;ERROR REGISTER
 2803  002762  000000                      .WORD   0               ;STATUS REGISTER
 2804  002764  000000                      .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
 2805  002766  000000                      .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
 2806  002770  000000                      .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
 2807  002772  000000                      .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
 2808  002774  000000                      .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
 2809  002776  000000                      .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
 2810  003000  000000                      .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
 2811  003002  000000                      .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
 2812  003004  000000                      .WORD   0               ;ECC POSITION INFORMATION
 2813  003006  000000                      .WORD   0               ;ECC PATTERN INFORMATION
 2814
 2815                                .SBTTL  TEMPORARY CONTROLLER REGISTER STORAGE
 2816
 2817  003010  000000              T.CS1:  .WORD   0               ;TEMPORARY STORAGE FOR COMMAND AND STATUS
 2818                                                              ;  REGISTER 1
 2819  003012  000000              T.CS2:  .WORD   0               ;TEMPORARY STORAGE FOR COMMAND AND STATUS
 2820                                                              ;  REGISTER 2
 2821  003014  000000              T.WCR:  .WORD   0               ;TEMPORARY STORAGE FOR WORD COUNT REGISTER
 2822  003016  000000              T.BA:   .WORD   0               ;TEMPORARY STORAGE FOR BUS ADDRESS REGISTER
 2823  003020  000000              T.DA:   .WORD   0               ;TEMPORARY STORAGE FOR DISK TRACK AND SECTOR
```

```
 2824  003022  000000                  T.DC:   .WORD   0               ;TEMPORARY STORAGE FOR DRIVE CYLINDER
 2825  003024  000000                  T.ASOF: .WORD   0               ;TEMPORARY STORAGE FOR ATTENTION SUMMARY
 2826                                                                  ;  AND OFFSET
 2827  003026  000000                  T.ER:   .WORD   0               ;TEMPORARY STORAGE FOR ERROR REGISTER
 2828  003030  000000                  T.DS:   .WORD   0               ;TEMPORARY STORAGE FOR DRIVE STATUS REGISTER
 2829  003032  000000                  T.MR1:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 1
 2830  003034  000000                  T.MR2:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 2
 2831  003036  000000                  T.MR3:  .WORD   0               ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 3
 2832  003040  000000                  T.POS:  .WORD   0               ;TEMPORARY STORAGE FOR ECC POSITION
 2833  003042  000000                  T.PAT:  .WORD   0               ;TEMPORARY STORAGE FOR ECC PATTERN
 2834  003044  000000                  T.DB:   .WORD   0               ;TEMPORARY STORAGE FOR DATA BUFFER REGISTER
 2835
 2836                                  .SBTTL  DRIVER PARAMERTERS
 2837
 2838  003046  177440                  RKBAS:  .WORD   177440          ;ADDRESS OF RK611 UNIBUS ADDRESS BLOCK
 2839  003050  000210                  RKVEC:  .WORD   210             ;ADDRESS OF R611 VECTOR
 2840  003052  000240                  RKPRI:  .WORD   PR5             ;RK611 INTERRUPT PRIORITY
 2841  003054  030250                  A.NORM: ERRFRE                  ;ADDRESS OF NORMAL RETURN FROM DRIVER
 2842  003056  031472                  A.ABNL: ERRHDL                  ;ADDRESS OF ABNORMAL RETURN FROM DRIVER
 2843  003060  030756                  A.CONT: CONERR                  ;ADDRESS OF CONTROLLER ERROR RETURN
 2844  003062  000000                  E.CONT: .WORD   0               ;CONTROLLER ERROR STATUS
 2845                                                                  ;  THIS LOCATION IS CLEARED WHEN EVERY COMMAND
 2846                                                                  ;  IS INITIATED.  IF A CONTROLLER ERROR
 2847                                                                  ;  OCCURS THE FOLLOWING BIT ASSIGNMENT IS
 2848                                                                  ;  USED:
 2849                                                                  ;
 2850          000001                  E.CCLF= BIT0                    ;CLEAR CONTROLLER DID NOT CLEAR ERROR
 2851          000002                  E.NOAT= BIT1                    ;NO ATTENTION IN ATTENTION SUMMARY REG
 2852          000004                  E.UATT= BIT2                    ;UNSOLICATED ATTENTION (SEQUENTIAL ONLY)
 2853          000010                  E.UDAT= BIT3                    ;UNEXPECTED DATA TYPE ERROR
 2854          000020                  E.CLAT= BIT4                    ;ATTENTION DID NOT RESET WITH CLEAR
 2855          000040                  E.SCLR= BIT5                    ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE
 2856                                                                  ;  ATTENTION
 2857          000100                  E.ILLD= BIT6                    ;ILLEGAL DRIVER COMMAND
 2858          000400                  E.DLT=  BIT8                    ;DATA LATE WHEN UNLOADING HEADER
 2859          001000                  E.CERR= BIT9                    ;CONTROLLER ERROR DURING DRIVER SERVICING
 2860          002000                  E.DPAR= BIT10                   ;DRIVE DETECTED PARITY ERROR
 2861          040000                  E.CMTO= BIT14                   ;CONTROLLER COMMAND TIME OUT (QUEUED ONLY)
 2862          100000                  E.MDS=  BIT15                   ;MULTIPLE DRIVE SELECT
 2863
 2864  003064  000000                  O.WAIT: .WORD   0               ;PARAMETER BLOCK OF THE DRIVE
 2865                                                                  ;WAITING FOR COMMAND COMPLETION
 2866  003066  000400                  W.MTIM: .WORD   400             ;LOOP COUNTER FOR MILISECOND SCAN OF DRIVE
 2867  003070  000400                  W.MILI: .WORD   400             ;16 MILISECOND TIME FOR PROGRAM
 2868                                                                  ;
 2869                                                                  ;       CPU     VALUE
 2870                                                                  ;       ---     -----
 2871                                                                  ;
 2872                                                                  ;       11/05   100
 2873                                                                  ;       11/10
 2874                                                                  ;       11/20
 2875                                                                  ;       11/34
 2876                                                                  ;       11/40
 2877                                                                  ;       11/45   400
 2878                                                                  ;       11/50
 2879                                                                  ;       11/70
```

DRIVER PARAMERTERS SEQ 0059

```
2880  003072  000300                  W.SEC:  .WORD   300             ;SECOND COUNT COUNT FOR ALL COMMANDS
2881                                                                  ;  EXCEPT START SPINDLE
2882  003074  003000                  W.8SEC: .WORD   3000            ;8 SECOND FOR DRIVE CYCLE DOWN
2883  003076  030000                  W.MIN:  .WORD   30000           ;MINUTE TIME FOR START SPINDLE
2884  003100  000000                  HDR.AD: .WORD   0               ;ADDRESS USED FOR READ ALL HEADERS
2885  003102  000000                  HDR.CT: .WORD   0               ;NUMBER OF HEADERS LEFT TO READ FOR READ
2886                                                                  ;  ALL HEADERS
2887  003104      000                 I.ISRL: .BYTE   0               ;INTERRUPT OR RELEASED COMMAND ISSUED
2888  003105      002     004     010 H.HEAD: .BYTE   2,4,10          ;HEAD DECODES
2889  003110      000                 W.TIME: .BYTE   0               ;DRIVES BEING WATCH-DOG TIMED
2890
2891                                  .SBTTL  INTERRUPT MASKS
2892
2893  003111      000                 INTMSK: .BYTE   0               ;INTERRUPT MASKS FOR DRIVE IN PARAMETER BLOCK
2894
2895                                  ;       INTERRUPT MASK TABLE
2896
2897  003112      001                 I.DRV:  .BYTE   1               ;INTERRUPT MASK FOR DRIVE 0
2898  003113      002                         .BYTE   2               ;INTERRUPT MASK FOR DRIVE 1
2899  003114      004                         .BYTE   4               ;INTERRUPT MASK FOR DRIVE 2
2900  003115      010                         .BYTE   10              ;INTERRUPT MASK FOR DRIVE 3
2901  003116      020                         .BYTE   20              ;INTERRUPT MASK FOR DRIVE 4
2902  003117      040                         .BYTE   40              ;INTERRUPT MASK FOR DRIVE 5
2903  003120      100                         .BYTE   100             ;INTERRUPT MASK FOR DRIVE 6
2904  003121      200                         .BYTE   200             ;INTERRUPT MASK FOR DRIVE 7
2905
2906                                  .SBTTL  PARAMETER BLOCK TABLE
2907
2908  003122  002640                  PBLKT:  PARM0                   ;ADDRESS OF PARAMETER BLOCK GIVEN WITH
2909                                                                  ;DRIVE CALL.  MUST BE LOADED INTO PBLKT
2910
2911
2912                                  .SBTTL  TIME FOR WATCH-DOG TIMER
2913
2914  003124  000000                  W.DRV:  .WORD   0               ;TIME FOR INSTRUCTION IN PARAMETER BLOCK
2915                                  .SBTTL  PROGRAM SPECIFIC RESERVED LOCATIONS
2916  003126      000                 MDFLAG: .BYTE   0               ;FLAG TO INDIC. DEFLT OR PARAM MODE
2917  003127      000                 TSTING: .BYTE   0               ;CURRENTLY RUNNING TESTS IF = 1
2918  003130      000                 DERCNT: .BYTE   0               ;DATA ERROR COUNT
2919  003131      000                 OPCOMP: .BYTE   0               ;OPERATION COMPLETE FLAG
2920  003132      000                 DONE:   .BYTE   0               ;DONE SWITCH
2921  003133      000                 TYPFMT: .BYTE   0               ;DRIVE TYPE & FORMAT CONTROL
2922  003134      000                 FORMAT: .BYTE   0               ;DRIVE FORMAT IN BIT 4 OF BYTE
2923  003135      000                 ERRCNT: .BYTE   0               ;ERROR COUNT
2924  003136      004                 ERRLMT: .BYTE   4               ;ERROR LIMIT
2925  003137      000                 DRVERS: .BYTE   0               ;ERROR COUNT FOR CURRENT DRIVE
2926  003140      000                 OPCONT: .BYTE   0               ;OPERATION CONTROL SWITCHES
2927  003141      000                 DRNAFG: .BYTE   0               ;=1 INDICATES DRIVE SIEZED BY OTHER PORT
2928  003142      000                 REISSU: .BYTE   0               ;DUAL-ACC FLAG TO RE-ISSUE COMMAND
2929  003143      000                 TSTTYP: .BYTE   0               ;OFFSET TEST IDENTIFIER
2930  003144      000                 DCEFLG: .BYTE   0               ;DATA CHECK ERROR FLAG
2931  003145      000                 WCEFLG: .BYTE   0               ;WRITE CHECK ERROR FLAG
2932  003146      000                 DLTFLG: .BYTE   0               ;DATA LATE ERROR FLAG
2933  003147      000                 DIF100: .BYTE   0               ;"CYL DIF = 100/200" FLAG
2934  003150      000                 NORTRY: .BYTE   0               ;"NO-RETRY" FLAG
2935  003151      000                 MEMABT: .BYTE   0               ;SET BYTE = 1 FOR NO ABORT
```

```
2936                                                                ;    ON MEMORY PARITY ERRORS
2937  003152     000                 HLPOVL: .BYTE   0              ;HELP FILE OVERLAID INDICATOR
2938  003153     000                 NOTYPE: .BYTE   0              ;INDICATES PROGRAM JUST LOADED IF 0
2939  003154     000                 LOGDRV: .BYTE   0              ;LOGICAL DRIVE NUMBER (00-17)
2940  003155     000                 PASSNO: .BYTE   0              ;PASS NUMBER (= 1 OR 2)
2941  003156     000                 OFSDIR: .BYTE   0              ;OFFSET DIRECTION FLAG (0=NEG)
2942  003157     000                 OFSCNT: .BYTE   0              ;OFFSET COUNT FOR ONE DIRECTION
2943          000001                 WHDSW=BIT0                     ;WRITE HEADER & DATA SWITCH
2944          000002                 VFHDSW=BIT1                    ;VERIFY HEADERS SWITCH
2945          000004                 WCDASW=BIT2                    ;WRITE CHECK DATA SWITCH
2946          000010                 RCDASW=BIT3                    ;READ CHECK DATA SWITCH
2947          000020                 OREQSW=BIT4                    ;OFFSET REQUIRED SWITCH
2948                                         .EVEN
2949
2950          000114                 MEMVEC=114                     ;MEMORY PARITY TRAP VECTOR
2951  003160  000000                 SAVWRD: .WORD   0              ;SCRATCH WORD
2952  003162  000400                 BSSOFT: .BLKW   ^D256          ;RECORD OF BAD SECTORS FROM SOFTWARE
2953  004162  000400                 BSFACT: .BLKW   ^D256          ;RECORD OF BAD SECTORS FROM FACTORY
2954  005162  000000  000000  000000 PRVCMD: .WORD   0,0,0,0,0,0     ;PREVIOUS COMMAND STORAGE
2955  005170  000000  000000  000000
2956  005176  000000  000000  000000 COMSTR: .WORD   0,0,0,0,0,0     ;CURRENT COMMAND STORAGE
2957  005204  000000  000000  000000
2958  005212  000102                 BUFF0:  .BLKW   ^D66           ;OUTPUT BUFFER 1
2959  005416  000000                 LOWOCT: .WORD   0              ;LOW 16 BITS OF CONVERTED BINARY NUMBER
2960  005420  000000                 HIGOCT: .WORD   0              ;HIGH BITS OF CONVERTED BINARY NO.
2961  005422  000000                 BUFPRT: .WORD   0              ;BUFFER POINTER
2962  005424  000000                 RECODE: .WORD   0              ;RECOVERY CODE WORD
2963  005426  000000                 ERRCOM: .WORD   0              ;ERROR COMMAND
2964  005430  000000                 DRIVE:  .WORD   0              ;NO.  OF  DRIVE IN USE
2965  005432  000000                 STALLS: .WORD   0              ;CURRENT NO. OF UNIT STALLS TO APPLY
2966  005434  000000                 CYLNDR: .WORD   0              ;CURRENT CYLINDER NUMBER
2967  005436  000000                 OFINUS: .WORD   0              ;OFFSET IN USE
2968  005440  000000                 TRACK:  .WORD   0              ;TRACK IN USE
2969  005442  000000                 INTCHR: .WORD   0              ;TTY INTERRUPT INPUT WORD
2970  005444  000000                 SCRACH: .WORD   0              ;ALL-PURPOSE SCRATCH WORD
2971  005446  000000                 PATRN:  .WORD   0              ;DATA PATTERN LIST WORD
2972  005450  000000                 PLOFST: .WORD   0              ;(+) OFFSET VALUE
2973  005452  000000                 NGOFST: .WORD   0              ;(-) OFFSET VALUE
2974  005454  000005                 SAVPRS: .BLKW   5              ;SAVE INITIAL PARAMS FOR XFER
2975  005466  000000                 WDSXFR: .WORD   0              ;NO. OF WORDS ACTUALLY XFERRED
2976  005470  000000                 FINCYL: .WORD   0              ;FINAL CYLINDER ADRS
2977  005472     000                 FINTRK: .BYTE   0              ;FINAL TRACK ADRS
2978  005473     000                 FINSEC: .BYTE   0              ;FINAL SECTOR ADRS
2979  005474  000000                 LASTWC: .WORD   0              ;ACTUAL FINAL WC
2980  005476  000000                 NEWON:  .WORD   0              ;NEW LOOK AT ONLINE DRIVES
2981  005500  000002                 MA:     .BLKW   2
2982  005504  000002                 PMA:    .BLKW   2              ;MEM. BUFFER ADDRESS
2983  005510  000000                 SUBSYS: .WORD   0              ;CURRENT SUBSYSTEM NAME STORAGE (ASCII)
2984  005512  000021                 DRVLST: .BLKW   17.            ;EACH WORD CONTAINS THE NAME OF A DRIVE
2985                                                                ;  TO BE TESTED. HIGH BYTE IS SUBSYSTEM
2986                                                                ;  NAME, LOW BYTE IS DRIVE NO. (ASCII).
2987                                                                ;  IF WORD = 0, DRIVE IS NOT TESTED.
2988  005554  000000                 DRVPTR: .WORD   0              ;DRIVE LIST POINTER
2989  005556  000010                 SCRLST: .BLKB   8.             ;SCRATCH DRIVE LIST
2990
2991  005566                         PAT00:
```

```
2992
2993
2994
2995
2996                                    ;TABLE A  -   BASIC READ/WRITE TEST SECTORS FOR RK06
2997  005566                            TABLEA:
2998  005566  000620  000000                    .WORD   620,0           ;DRIVE  0
2999  005572  000620  000002                    .WORD   620,2           ;       1
3000  005576  000620  000004                    .WORD   620,4           ;       2
3001  005602  000620  000006                    .WORD   620,6           ;       3
3002  005606  000620  000010                    .WORD   620,10          ;       4
3003  005612  000620  000012                    .WORD   620,12          ;       5
3004  005616  000620  000014                    .WORD   620,14          ;       6
3005  005622  000620  000016                    .WORD   620,16          ;       7
3006  005626  000622  000000                    .WORD   622,0           ;       10
3007  005632  000622  000002                    .WORD   622,2           ;       11
3008  005636  000622  000004                    .WORD   622,4           ;       12
3009  005642  000622  000006                    .WORD   622,6           ;       13
3010  005646  000622  000010                    .WORD   622,10          ;       14
3011  005652  000622  000012                    .WORD   622,12          ;       15
3012  005656  000622  000014                    .WORD   622,14          ;       16
3013  005662  000622  000016                    .WORD   622,16          ;       17
3014
3015                                    ;TABLE A -  CORRESPONDING CYL NUMBERS FOR THE RK07
3016
3017  005666  001444  000000            TABL7A: .WORD   1444,0          ;DRV 0
3018  005672  001444  000002                    .WORD   1444,2          ;1
3019  005676  001444  000004                    .WORD   1444,4          ;2
3020  005702  001444  000006                    .WORD   1444,6          ;3
3021  005706  001444  000010                    .WORD   1444,10         ;4
3022  005712  001444  000012                    .WORD   1444,12         ;5
3023  005716  001444  000014                    .WORD   1444,14         ;6
3024  005722  001444  000016                    .WORD   1444,16         ;7
3025  005726  001446  000000                    .WORD   1446,0          ;10
3026  005732  001446  000002                    .WORD   1446,2          ;11
3027  005736  001446  000004                    .WORD   1446,4          ;12
3028  005742  001446  000006                    .WORD   1446,6          ;13
3029  005746  001446  000010                    .WORD   1446,10         ;14
3030  005752  001446  000012                    .WORD   1446,12         ;15
3031  005756  001446  000014                    .WORD   1446,14         ;16
3032  005762  001446  000016                    .WORD   1446,16         ;17
3033
3034
3035
3036                                    ;TABLE B  -   WORST CASE DATA PATTERN
3037  005766                            TABLEB:
3038  005766  072307                            072307
3039  005770  135143                            135143
3040  005772  156461                            156461
3041  005774  167230                            167230
3042  005776  073514                            073514
3043  006000  035646                            035646
3044  006002  016723                            016723
3045  006004  107351                            107351
3046  006006  143564                            143564
3047  006010  061672                            061672
```

```
  3048  006012  030735                          030735
  3049  006014  114356                          114356
  3050  006016  046167                          046167
  3051  006020  123073                          123073
  3052  006022  151453                          151453
  3053  006024  164616                          164616
  3054
  3055
  3056
  3057                                ;TABLE OF STARTING CYLINDERS FOR ALL OVWRT AND COMPAT CYL BLKS FOR RK06
  3058  006026  000000  000060        TABLEC: .WORD   0,60            ;FOR CURRENT ZONE 1
  3059  006032  000100  000160                .WORD   100,160         ;                 2
  3060  006036  000200  000260                .WORD   200,260         ;                 3
  3061  006042  000300  000360                .WORD   300,360         ;                 4
  3062  006046  000400  000460                .WORD   400,460         ;                 5
  3063  006052  000500  000560                .WORD   500,560         ;                 6
  3064  006056  000600                        .WORD   600             ;                 7
  3065
  3066                                ;TABLE C  CORRESPONDING CYL NUMBERS FOR RK07
  3067
  3068  006060  000000  000160        TABL7C: .WORD   0,160           ;CURRENT ZONE 1
  3069  006064  000200  000360                .WORD   200,360         ;2
  3070  006070  000400  000560                .WORD   400,560         ;3
  3071  006074  000600  000760                .WORD   600,760         ;4
  3072  006100  001000  001160                .WORD   1000,1160       ;5
  3073  006104  001200  001360                .WORD   1200,1360       ;6
  3074  006110  001400                        .WORD   1400            ;7
  3075
  3076
  3077
  3078                                ;CYLINDER BLOCK LAYOUT TABLE
  3079  006112  000020                TABLED: .BLKB   16.             ;CYLINDER 0 IN BLK
  3080  006132  000020                        .BLKB   16.             ;         1
  3081  006152  000020                        .BLKB   16.             ;         2
  3082  006172  000020                        .BLKB   16.             ;         3
  3083  006212  000020                        .BLKB   16.             ;         4
  3084  006232  000020                        .BLKB   16.             ;         5
  3085  006252  000020                        .BLKB   16.             ;         6
  3086  006272  000020                        .BLKB   16.             ;         7
  3087  006312  000020                        .BLKB   16.             ;        10
  3088  006332  000020                        .BLKB   16.             ;        11
  3089  006352  000020                        .BLKB   16.             ;        12
  3090  006372  000020                        .BLKB   16.             ;        13
  3091  006412  000020                        .BLKB   16.             ;        14
  3092  006432  000020                        .BLKB   16.             ;        15
  3093  006452  000020                        .BLKB   16.             ;        16
  3094  006472  000020                        .BLKB   16.             ;        17
  3095
  3096
  3097
  3098                                ;TABLE G - PSEUDO-RANDOM DATA PATTERN
  3099  006512                        TABLEG:
  3100  006512  040135                        040135
  3101  006514  177070                        177070
  3102  006516  070414                        070414
  3103  006520  064531                        064531
```

```
  3104  006522  174473                          174473
  3105  006524  062422                          062422
  3106  006526  114352                          114352
  3107  006530  036620                          036620
  3108  006532  010031                          010031
  3109  006534  052336                          052336
  3110  006536  017310                          017310
  3111  006540  011347                          011347
  3112  006542  102367                          102367
  3113  006544  152567                          152567
  3114  006546  001246                          001246
  3115  006550  160073                          160073
  3116
  3117
  3118
  3119                                  ;LIST OF STARTING DRIVE NUMBERS, TO GENERATE TABLED
  3120  006552     000     001     003  STDLST: .BYTE   0,1,3,6,12,17,5,14,4,15,7,2,16,13,11,10
  3121  006555     006     012     017
  3122  006560     005     014     004
  3123  006563     015     007     002
  3124  006566     016     013     011
  3125  006571     010
  3126
  3127
  3128
  3129                                  ;LIST OF USEABLE SECTORS IN EACH CYLINDER BLOCK
  3130  006572     000     001     002  SECLST: .BYTE   0,1,2,3,5,6,7,10,12,13,14,15,17,20,21,22
  3131  006575     003     005     006
  3132  006600     007     010     012
  3133  006603     013     014     015
  3134  006606     017     020     021
  3135  006611     022
  3136
  3137
  3138
  3139                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 0 FOR NEG OFST
  3140  006612  000020                    NOSCR0: .BLKW   16.
  3141
  3142                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 1 FOR NEG OFST
  3143  006652  000020                    NOSCR1: .BLKW   16.
  3144
  3145                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 2 FOR NEG OFST
  3146  006712  000020                    NOSCR2: .BLKW   16.
  3147
  3148
  3149
  3150                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 0 FOR POS OFST
  3151  006752  000020                    POSCR0: .BLKW   16.
  3152
  3153                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 1 FOR POS OFST
  3154  007012  000020                    POSCR1: .BLKW   16.
  3155
  3156                                  ;TABLE OF OVERWRITE TEST SCORES FOR TRACK 2 FOR POS OFST
  3157  007052  000020                    POSCR2: .BLKW   16.
  3158
  3159
```

```
3160
3161                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 0 FOR NEG OFST
3162  007112  000020                    NCSCR0: .BLKW   16.
3163
3164                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 1 FOR NEG OFST
3165  007152  000020                    NCSCR1: .BLKW   16.
3166
3167                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 2 FOR NEG OFST
3168  007212  000020                    NCSCR2: .BLKW   16.
3169
3170
3171
3172                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 0 FOR POS OFST
3173  007252  000020                    PCSCR0: .BLKW   16.
3174
3175                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 1 FOR POS OFST
3176  007312  000020                    PCSCR1. .BLKW   16.
3177
3178                                    ;TABLE OF DATA COMPATIBILITY TEST SCORES FOR TRACK 2 FOR POS OFST
3179  007352  000020                    PCSCR2: .BLKW   16.
3180
3181
3182
3183
3184                                    ;SELF-TEST SCORE FOR TRACK 0 FOR NEG OFST
3185  007412  000000                    NSSCR0: .WORD   0
3186
3187                                    ;SELF-TEST SCORE FOR TRACK 1 FOR NEG OFST
3188  007414  000000                    NSSCR1: .WORD   0
3189
3190                                    ;SELF-TEST SCORE FOR TRACK 2 FOR NEG OFST
3191  007416  000000                    NSSCR2: .WORD   0
3192
3193
3194
3195                                    ;SELF-TEST SCORE FOR TRACK 0 FOR POS OFST
3196  007420  000000                    PSSCR0: .WORD   0
3197
3198                                    ;SELF-TEST SCORE FOR TRACK 1 FOR POS OFST
3199  007422  000000                    PSSCR1: .WORD   0
3200
3201                                    ;SELF-TEST SCORE FOR TRACK 2 FOR POS OFST
3202  007424  000000                    PSSCR2: .WORD   0
3203
3204
3205
3206                                    ;AVERAGE OF OVRWRT SCORES OVER ALL SURFACES
3207  007426  000000                    OVRAVG: .WORD   0
3208
3209                                    ;AVERAGE OF READ SCORES OVER ALL SURFACES
3210  007430  000000                    COMAVG: .WORD   0
3211
3212
3213
3214                                    ;SCRATCH TABLE OF SCORES FOR TRACK 0 FOR NEG OFST
3215  007432  000020                    NSCOR0: .BLKB   16.
```

```
3216
3217                                    ;SCRATCH TABLE OF SCORES FOR TRACK 1 FOR NEG OFST
3218  007452  000020                    NSCOR1: .BLKB   16.
3219
3220                                    ;SCRATCH TABLE OF SCORES FOR TRACK 2 FOR NEG OFST
3221  007472  000020                    NSCOR2: .BLKB   16.
3222
3223
3224
3225                                    ;SCRATCH TABLE OF SCORES FOR TRACK 0 FOR POS OFST
3226  007512  000020                    PSCOR0: .BLKB   16.
3227
3228                                    ;SCRATCH TABLE OF SCORES FOR TRACK 1 FOR POS OFST
3229  007532  000020                    PSCOR1: .BLKB   16.
3230
3231                                    ;SCRATCH TABLE OF SCORES FOR TRACK 2 FOR POS OFST
3232  007552  000020                    PSCOR2: .BLKB   16.
3233
3234
3235
3236          000002                    LSTTRK=2                        ;LAST TRACK  = 2
3237          000002                    BSERR=BIT1                      ;BSE ERROR
3238          000004                    HVRCER=BIT2                     ;HVRC ERROR
3239          000010                    OPIERR=BIT3                     ;OPI ERROR
3240          000020                    DCKERR=BIT4                     ;DATA CHECK ERROR
3241          000040                    ECCNC=BIT5                      ;ECC NON-CORRECTABLE
3242          000100                    WCERR=BIT6                      ;WRITE CHECK ERROR
3243          000200                    ABORT=BIT7                      ;ABORT
3244          000400                    LEV2ER=BIT8                     ;LEVEL TWO ERROR
3245          001000                    BADSEC=BIT9                     ;BAD SECTOR FLAG
3246          002000                    TWOTOS=BIT10                    ;TWO TIME OUTS
3247          004000                    RCLREQ=BIT11                    ;RECALIBRATE REQUIRED
3248          010000                    DRNAVL=BIT12                    ;DRIVE NOT RELSD BY OTHER PORT
3249          100000                    ANYDER=BIT15                    ;ANY ERROR DETECTED FLAG
3250
3251  007572  005015  041412  051132  DZR6Q:   .ASCIZ  <15><12><12>/CZR6QC0 RK6 DR CPT PROG/<15><12><12>
3252  007600  050466  030103  051040
3253  007606  033113  042040  020122
3254  007614  050103  020124  051120
3255  007622  043517  005015  000012
3256  007630  051012  030113  026466  RKBADR: .ASCIZ  <12>/RK06-07 BUS ADR = /
3257  007636  033460  041040  051525
3258  007644  040440  051104  036440
3259  007652  000040
3260  007654  045522  033060  030055  RKVADR: .ASCIZ  /RK06-07 VEC ADR = /
3261  007662  020067  042526  020103
3262  007670  042101  020122  020075
3263  007676     000
3264  007677     122  030113  026466  RKPRTY: .ASCIZ  /RK06-07 PRIORITY =/
3265  007704  033460  050040  044522
3266  007712  051117  052111  020131
3267  007720  000075
3268  007722  052012  050131  020105  TYPSYS: .ASCIZ  <12>/TYPE NAME OF NEXT SUBSYS TO TEST (A-P) : /
3269  007730  040516  042515  047440
3270  007736  020106  042516  052130
3271  007744  051440  041125  054523
```

```
  3272  007752  020123  047524  052040
  3273  007760  051505  020124  040450
  3274  007766  050055  020051  020072
  3275  007774     000
  3276  007775     012  051104  053111  ASKTYP: .ASCIZ  <12>/DRIVE TYPE  6<CR> FOR RK06, 7<CR> FOR RK07: /
  3277  010002  020105  054524  042520
  3278  010010  020040  036066  051103
  3279  010016  020076  047506  020122
  3280  010024  045522  033060  020054
  3281  010032  036067  051103  020076
  3282  010040  047506  020122  045522
  3283  010046  033460  020072     000
  3284  010053     123  041125  054523  SYSDRV: .ASCIZ  /SUBSYS   DRIVE(S) = /
  3285  010060  020123  020040  051104
  3286  010066  053111  024105  024523
  3287  010074  036440  000040
  3288  010100  044527  046114  052040  WILTST: .ASCIZ  /WILL TEST DRIVE(S) /
  3289  010106  051505  020124  051104
  3290  010114  053111  024105  024523
  3291  010122  000040
  3292  010124  047440  020116  052523  ONSUBS: .ASCIZ  / ON SUBSYS  /<15><12><12>
  3293  010132  051502  051531  020040
  3294  010140  005015  000012
  3295  010144  051511  052040  042510  ANOTHR: .ASCIZ  /IS THERE ANOTHER SUBSYS (Y OR N <CR>) ? /
  3296  010152  042522  040440  047516
  3297  010160  044124  051105  051440
  3298  010166  041125  054523  020123
  3299  010174  054450  047440  020122
  3300  010202  020116  041474  037122
  3301  010210  020051  020077     000
  3302  010215     012  047515  042522  DROVFL: .ASCIZ  <12>/MORE THAN 16 DRIVES ENTERED !/<15><12><12>
  3303  010222  052040  040510  020116
  3304  010230  033061  042040  044522
  3305  010236  042526  020123  047105
  3306  010244  042524  042522  020104
  3307  010252  006441  005012     000
  3308  010257     012  025052  051440  SRPASS: .ASCIZ  <12>/** STARTING PASS   **/<15><12>
  3309  010264  040524  052122  047111
  3310  010272  020107  040520  051523
  3311  010300  020040  025040  006452
  3312  010306  000012
  3313  010310  046412  052517  052116  MNTPAK: .ASCIZ  <12>/MOUNT PACK ON DRIVE    AND LOAD./<15><12>
  3314  010316  050040  041501  020113
  3315  010324  047117  042040  044522
  3316  010332  042526  020040  020040
  3317  010340  047101  020104  047514
  3318  010346  042101  006456  000012
  3319  010354  052412  046116  040517  DISPAK: .ASCIZ  <12>/UNLOAD DRIVE    AND REMOVE PACK./<15><12>
  3320  010362  020104  051104  053111
  3321  010370  020105  020040  040440
  3322  010376  042116  051040  046505
  3323  010404  053117  020105  040520
  3324  010412  045503  006456  000012
  3325  010420  051412  040524  052122  STR204: .ASCIZ  <12>/START AT ADR 204 FOR PASS 1 ON SUBSYS  ./<15><12><12>
  3326  010426  040440  020124  042101
  3327  010434  020122  030062  020064
```

```
  3328  010442  047506  020122  040520
  3329  010450  051523  030440  047440
  3330  010456  020116  052523  051502
  3331  010464  051531  020040  006456
  3332  010472  005012     000
  3333  010475     012  052123  051101  STR220: .ASCIZ  <12>/START AT ADR 220 FOR PASS 2 ON SUBSYS  ./<15><12><12>
  3334  010502  020124  052101  040440
  3335  010510  051104  031040  030062
  3336  010516  043040  051117  050040
  3337  010524  051501  020123  020062
  3338  010532  047117  051440  041125
  3339  010540  054523  020123  027040
  3340  010546  005015  000012
  3341  010552  054524  042520  051040  TYPRWN: .ASCIZ  /TYPE R<CR> WHEN /
  3342  010560  041474  037122  053440
  3343  010566  042510  020116     000
  3344  010573     104  044522  042526  DRIRDY: .ASCIZ  /DRIVE READY : /
  3345  010600  051040  040505  054504
  3346  010606  035040  000040
  3347  010612  047504  042516  035040  DRIDON: .ASCIZ  /DONE : /
  3348  010620  000040
  3349  010622  047516  042040  044522  NODRLS: .ASCIZ  /NO DRIVES LISTED FOR THIS SUBSYS !/<15><12>
  3350  010630  042526  020123  044514
  3351  010636  052123  042105  043040
  3352  010644  051117  052040  044510
  3353  010652  020123  052523  051502
  3354  010660  051531  020440  005015
  3355  010666     000
  3356  010667     015  051412  047503  RESLTS: .ASCII  <15><12>/SCORES FOR DRIVE     (0-12 DEC.) :/<15><12><12>
  3357  010674  042522  020123  047506
  3358  010702  020122  051104  053111
  3359  010710  020105  020040  020040
  3360  010716  030050  030455  020062
  3361  010724  042504  027103  020051
  3362  010732  006472  005012
  3363  010736  051124  041501  004513          .ASCII  /TRACK  DRIVE   OVRWRT  OVRWRT  READ    READ/<15><12>
  3364  010744  051104  053111  004505
  3365  010752  053117  053522  052122
  3366  010760  047411  051126  051127
  3367  010766  004524  042522  042101
  3368  010774  051011  040505  006504
  3369  011002     012
  3370  011003     116  046525  051011          .ASCII  /NUM    READ    SCORE   SCORE   SCORE   SCORE/<15><12>
  3371  011010  040505  004504  041523
  3372  011016  051117  004505  041523
  3373  011024  051117  004505  041523
  3374  011032  051117  004505  041523
  3375  011040  051117  006505     012
  3376  011045     011  047411  051506          .ASCIZ  /               OFST-   OFST+   OFST-   OFST+/<15><12><12>
  3377  011052  026524  047411  051506
  3378  011060  025524  047411  051506
  3379  011066  026524  047411  051506
  3380  011074  025524  005015  000012
  3381  011102  047440  051126  051127  OAVERG: .ASCIZ  / OVRWRT AVG = /
  3382  011110  020124  053101  020107
  3383  011116  020075     000
```

```
 3384  011121     040  042522  042101  RAVERG: .ASCIZ  / READ AVG   = /
 3385  011126  040440  043526  020040
 3386  011134  036440  000040
 3387  011140  005015  025012  020052  ENDTST: .ASCIZ  <15><12><12>/**  END OF TESTING  **/<15><12>
 3388  011146  042440  042116  047440
 3389  011154  020106  042524  052123
 3390  011162  047111  020107  025040
 3391  011170  006452  000012
 3392  011174  020040  000011          TRKBUF: .ASCIZ  /       /
 3393  011200  020040  004440     000  DRVBUF: .ASCIZ  /       /
 3394  011205     123  046105  004506  SELF:   .ASCIZ  /SELF   /
 3395  011212     000
 3396  011213     011     000          TAB:    .ASCIZ  /       /
 3397  011215     123  051127  036440  SWRMSG: .ASCIZ  /SWR = /
 3398  011222  000040
 3399  011224  020040  042516  020127  NEWMSG: .ASCIZ  /  NEW = /
 3400  011232  020075     000
 3401  011235     015  042012  044522  DRVSEQ: .ASCIZ  <15><12>/DRIVE(S) = /
 3402  011242  042526  051450  020051
 3403  011250  020075     000
 3404  011253     104  044522  042526  BADDRV: .ASCIZ  /DRIVE   /
 3405  011260  020040  000040
 3406  011264  047516  026516  054105  NXDRIV: .ASCIZ  /NON-EXISTENT/<15><12>
 3407  011272  051511  042524  052116
 3408  011300  005015     000
 3409  011303     116  052117  051040  NTREDY: .ASCIZ  /NOT READY/<15><12>
 3410  011310  040505  054504  005015
 3411  011316     000
 3412  011317     127  044522  042524  WRTLOK: .ASCIZ  /WRITE-LOCKED/<15><12>
 3413  011324  046055  041517  042513
 3414  011332  006504  000012
 3415  011336  047514  042101  042105  ALNPAK: .ASCIZ  /LOADED WITH ALIGN PACK/<15><12>
 3416  011344  053440  052111  020110
 3417  011352  046101  043511  020116
 3418  011360  040520  045503  005015
 3419  011366     000
 3420  011367     111  020106  054130  REPLPK: .ASCIZ  /IF XXDP PACK ON DRV 0, TYPE "Y <CR>", &  REPLACE IT : /
 3421  011374  050104  050040  041501
 3422  011402  020113  047117  042040
 3423  011410  053122  030040  020054
 3424  011416  054524  042520  021040
 3425  011424  020131  041474  037122
 3426  011432  026042  023040  020040
 3427  011440  042522  046120  041501
 3428  011446  020105  052111  035040
 3429  011454  000040
 3430  011456  005015  025012  020052  NODRTS: .ASCII  <15><12><12>/** NO DRIVES TO TEST/<15><12>
 3431  011464  047516  042040  044522
 3432  011472  042526  020123  047524
 3433  011500  052040  051505  006524
 3434  011506     012
 3435  011507     120  042522  051523  CNTRDY: .ASCIZ  /PRESS "CONT" WHEN RDY/<15><12>
 3436  011514  021040  047503  052116
 3437  011522  020042  044127  047105
 3438  011530  051040  054504  005015
 3439  011536     000
```

```
 3440  011537     110  046101  020124  HLTRQD: .ASCIZ  /HALT REQUESTED/<15><12>
 3441  011544  042522  052521  051505
 3442  011552  042524  006504  000012
 3443  011560  051104  053111  020105  DROXDP: .ASCIZ  /DRIVE 0 IS LOAD MEDIUM/<15><12>
 3444  011566  020060  051511  046040
 3445  011574  040517  020104  042515
 3446  011602  044504  046525  005015
 3447  011610     000
 3448  011611     015  025012  020052  DROPDR: .ASCIZ  <15><12>/**  DROPPING DRIVE - 20 ERRORS/<15><12>
 3449  011616  042040  047522  050120
 3450  011624  047111  020107  051104
 3451  011632  053111  020105  020055
 3452  011640  030062  042440  051122
 3453  011646  051117  006523  000012
 3454  011654  005015  052012  051505  TSTDRN: .ASCII  <15><12><12>/TESTING DRIVE /
 3455  011662  044524  043516  042040
 3456  011670  044522  042526     040
 3457  011675     040  005015     000  DRVNO:  .ASCIZ  / /<15><12>
 3458  011701     104  044522  042526  DRIV:   .ASCIZ  /DRIVE /
 3459  011706  000040
 3460  011710  040503  052122  020056  CART:   .ASCIZ  /CART. /
 3461  011716     000
 3462  011717     123  051105  020056  SERNM:  .ASCIZ  /SER. NO.  /
 3463  011724  047516  020056  000040
 3464  011732  005015  040506  052103  FACTBS: .ASCIZ  <15><12>/FACTORY /
 3465  011740  051117  020131     000
 3466  011745     012  047523  052106  SOFTBS: .ASCII  <12>/SOFTWARE /
 3467  011752  040527  042522     040
 3468  011757     102  042101  051440  BDSECT: .ASCIZ  /BAD SECTORS :/<15><12>
 3469  011764  041505  047524  051522
 3470  011772  035040  005015     000
 3471  011777     040  047040  047117  NOFALS: .ASCIZ  /  NONE/
 3472  012004  000105
 3473
 3474  012006  025052  020040  040503  BAD632: .ASCIZ  /**  CANNOT READ BAD SECTOR TRACK!/<15><12>
 3475  012014  047116  052117  051040
 3476  012022  040505  020104  040502
 3477  012030  020104  042523  052103
 3478  012036  051117  052040  040522
 3479  012044  045503  006441  000012
 3480  012052  041536  005015     000  CNTRLC: .ASCIZ  /^C/<15><12>
 3481  012057     136  006532  000012  CNTRLZ: .ASCIZ  /^Z/<15><12>
 3482  012064  051136  005015     000  CNTRLR: .ASCIZ  /^R/<15><12>
 3483  012071     136  006525  000012  CNTRLU: .ASCIZ  /^U/<15><12>
 3484  012076  043536  005015     000  CNTRLG: .ASCIZ  /^G/<15><12>
 3485  012103     015  005012     000  CR2LF:  .ASCIZ  <15><12><12>
 3486  012107     134     000          BKSLSH: .ASCIZ  /\/
 3487  012111     054     000          COMMA:  .ASCIZ  /,/
 3488  012113     040     040          SPACE6: .ASCII  /  /
 3489  012115     040                  SPACE4: .ASCII  / /
 3490  012116     040                  SPACE3: .ASCII  / /
 3491  012117     040                  SPACE2: .ASCII  / /
 3492  012120  000040                  SPACE1: .ASCIZ  / /
 3493                                          .EVEN
 3494  012122  020040  000075          PRMBUF: .ASCIZ  /  =/
 3495  012126  047527  042122  000040  WORDSP: .ASCIZ  /WORD /
```

PROGRAM SPECIFIC RESERVED LOCATIONS SEQ 0070

```
3496  012134  036440  000040          EQUALS: .ASCIZ  / = /
3497  012140  020052     000          PROMPT: .ASCIZ  /* /
3498  012143     076  000040          PRMPSP: .ASCIZ  /> /
3499
3500                                          .EVEN
3501
3502
3503
3504
3505  012146  105037  003126          DFSTRT: CLRB    MDFLAG          ;SET FLAG FOR 200 START
3506  012152  000403                          BR      P1SET           ;CONTINUE
3507
3508  012154  112737  000001  003126  P1STRT: MOVB    #1,MDFLAG       ;SET FLAG FOR 204 START
3509  012162  112737  000061  003155  P1SET:  MOVB    #'1,PASSNO      ;SET PASS NO. = 1
3510  012170  000406                          BR      CMSTRT          ;CONTINUE
3511
3512  012172  112737  000002  003126  P2STRT: MOVB    #2,MDFLAG       ;SET FLAG FOR 220 START
3513  012200  112737  000062  003155          MOVB    #'2,PASSNO      ;SET PASS NO. = 2
3514
3515  012206  012737  000340  177776  CMSTRT: MOV     #PR7,@#PS       ;BLOCK ALL INTERRUPTS
3516                                  .SBTTL  INITIALIZE THE COMMON TAGS
3517                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3518  012214  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3519  012220  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3520  012222  022706  001140                  CMP     #SWR,R6 ;;DONE?
3521  012226  001374                          BNE     .-6             ;;LOOP BACK IF NO
3522  012230  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3523                                  ;;INITIALIZE A FEW VECTORS
3524  012234  012737  044574  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
3525  012242  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
3526  012250  012737  044070  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
3527  012256  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
3528  012264  012737  045306  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3529  012272  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
3530  012300  012737  044440  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3531  012306  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3532  012314  013737  017220  017212          MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
3533  012322  005037  001316                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
3534  012326  112737  000001  001115          MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
3535  012334  012737  012334  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3536  012342  012737  012342  001110          MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
3537                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3538                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3539  012350  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3540  012354  012737  012410  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3541  012362  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
3542  012370  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3543  012376  022777  177777  166534          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3544  012404  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3545                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
3546  012406  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
3547  012410  012716  012416          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3548  012414  000002                          RTI
3549  012416  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3550  012424  012737  000174  001142          MOV     #DISPREG,DISPLAY
3551  012432  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
```

```
                                                CLR     $PASS           ;;CLEAR PASS COUNT
012436  005037  001336                          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
012442  132737  000200  001351                  BEQ     67$             ;;YES,USE NON-APT SWITCH
012450  001403                                  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
012452  012737  001352  001140          67$:
012460                                          RESET                   ;CLEAR THE UNIBUS
012460  000005                                  MOV     #6,@#ERRVEC     ;SET TIME-OUT VECTOR
012462  012737  000006  000004                  CLR     @#ERRVEC+2
012470  005037  000006                          CLRB    TSTTYP          ;CLEAR OFFSET TEST IDENTIFIER
012474  105037  003143                          CLRB    DRVERS          ;CLEAR ERROR COUNT FOR CURRENT DRIVE
012500  105037  003137                          CLRB    ERRCNT          ;CLEAR ERROR RETRY COUNT
012504  105037  003135                          CLRB    $TSTNM          ;SET TEST NO. = 0
012510  105037  001102                          MOVB    #1,HLPOVL       ;SET HLP FILE OVLD INDICTR
012514  112737  000001  003152                  TYPE    ,DZR6Q          ;TYPE PROGRAM I.D.
012522  104401  007572                          TSTB    NOTYPE          ;SEE IF OPERATOR NOTE SHOULD BE TYPED
012526  105737  003153                          BNE     40$             ;BR IF NOT
012532  001004                                  TYPE    ,NOTMSG         ;TYPE OPERATOR NOTE
012534  104401  054330                          INCB    NOTYPE          ;SET FLAG FOR NEXT TIME
012540  105237  003153                  40$:    MOV     #KBDHDL,@#TKVEC ;LOAD VECTOR FOR TTY KBD
012544  012737  020604  000060                  MOV     #PR4,@#TKVEC+2  ;SET KBD PRIORITY = 4
012552  012737  000200  000062                  MOV     RKVEC,R1        ;ADDR. OF RK06 VECTOR STORAGE
012560  013701  003050                          MOV     #I.INTR,(R1)+   ;SET IT TO RK06 HANDLER
012564  012721  034710                          MOV     RKPRI,(R1)      ;SET RK06 PRIORITY
012570  013711  003052                          CLRB    TSTING          ;CLEAR "RUNNING TESTS" FLAG
012574  105037  003127                          MOV     #PR0,@#PS       ;ALLOW ALL INTERRUPTS AGAIN
012600  012737  000000  177776                  MOV     #PRVCMD,R1      ;ZERO OUT PREVIOUS COMMAND
012606  012701  005162                          MOV     #6,R0           ;SET COUNTER
012612  012700  000006                  42$:    CLR     (R1)+           ;ZERO A WORD
012616  005021                                  DEC     R0
012620  005300                                  BNE     42$             ;BR IF NOT DONE YET
012622  001375                                  CLR     STALLS          ;DON'T ALLOW STALLS YET
012624  005037  005432                          JSR     PC,GTSWRG       ;OPEN SOFTWARE SWR FOR MODIFICATION
012630  004737  021106                  44$:    MOV     #176543,$HINUM  ;INIT. PSEUDO-RANDOM NOS.
012634  012737  176543  042336                  MOV     #123456,$LONUM
012642  012737  123456  042340                  JSR     PC,GETTYP       ;SEE IF RK06 OR RK07 & SETUP
012650  004737  024154                          TSTB    MDFLAG          ;SEE IF DEFAULT MODE
012654  105737  003126                          BNE     ALLSYS          ;BR IF NOT DEFAULT MODE
012660  001034                                  MOV     $VECT1,R0       ;GET APT RK06 VECTOR AND PRTY
012662  013700  001400                          MOV     $BASE,RKBAS     ;GET APT RK06 BASE ADDRESS
012666  013737  001404  003046                  BITB    #BIT7,$ENVM     ;SEE IF NO SIZING
012674  132737  000200  001351                  BNE     18$             ;BR IF NO SIZING
012702  001005                                  MOV     #AVECT1,R0      ;GET DEFAULT VECTOR AND PRIORITY
012704  012700  120210                          MOV     #ABASE,RKBAS    ;GET DEFAULT BASE ADDRESS
012710  012737  177440  003046          18$:    MOVB    R0,RKVEC        ;STORE VECTOR
012716  110037  003050                          CLRB    RKVEC+1         ;CLEAR HI BYTE
012722  105037  003051                          SWAB    R0              ;GET PRTY INTO BITS 5-7
012726  000300                                  BIC     #177437,R0      ;CLEAR OTHER BITS
012730  042700  177437                          MOV     R0,RKPRI        ;STORE RK06 PRIORITY
012734  010037  003052                          MOVB    #'A,SUBSYS      ;SET SUBSYSTEM NAME = A
012740  112737  000101  005510                  JMP     DFLTDR          ;GO CHECK ALL DRIVES
012746  000137  013744


                                        ;;**************************************************************
                                        ;*
```

```
                                        .SBTTL  TEST 0     PARAMETER INPUT ROUTINES
                                        ;*
                                        ;;*****************************************************************
                                        ;READ AND STORE ALL SUBSYSTEMS AND DRIVES TO TEST
012752  104401  001325          ALLSYS: TYPE    ,$CRLF                  ;TYPE <CR>,<LF>
012756  105037  001102                  CLRB    $TSTNM                  ;SET TEST NO. = 0
012762  112703  000101                  MOVB    #'A,R3                  ;INIT SUBSYS INDICATOR
012766  012701  005512                  MOV     #DRVLST,R1              ;GET DRIVE LIST ADDRESS
012772  012700  000021                  MOV     #17.,R0                 ;CLEAR OUT THE DRIVE LIST
012776  005021                  2$:     CLR     (R1)+
013000  005300                          DEC     R0
013002  001375                          BNE     2$
013004  012704  005512                  MOV     #DRVLST,R4              ;GET STARTING ADRS OF DRIVE LIST
                                        ;ASK FOR DRIVES TO TEST ON THIS SUBSYSTEM
013010  110337  010062          ALLSY1: MOVB    R3,SYSDRV+7             ;SET SUBSYS NAME FOR PRINTOUT
013014  104401  010053          6$:     TYPE    ,SYSDRV                 ;ASK FOR DRIVES ON THIS SUBSYSTEM
013020  004737  023402                  JSR     PC,RDCHRS               ;READ INPUT STRING
013024  012752                          ALLSYS                          ;(^C) RETURN ADDRESS
013026  012752                          ALLSYS                          ;(^Z) RETURN ADDRESS
013030  013014                          6$                              ;(^U) RETURN ADDRESS
013032  005700                          TST     R0                      ;SEE IF NULL INPUT
013034  001767                          BEQ     6$                      ;BR TO ASK AGAIN
013036  022700  000017                  CMP     #15.,R0                 ;SEE IF TOO MANY CHARS TYPED
013042  002005                          BGE     12$                     ;BR IF NOT TOO MANY
013044  104401  005212          10$:    TYPE    ,BUFF0                  ;ECHO BAD INPUT
013050  104401  001324                  TYPE    ,$QUES                  ;TYPE <?> AND <CR>,<LF>
013054  000757                          BR      6$                      ;GO ASK AGAIN
013056  005001                  12$:    CLR     R1                      ;CLEAR CHAR POINTER
013060  126127  005212  000060  14$:    CMPB    BUFF0(R1),#'0           ;SEE IF DRIVE NO. < 0
013066  103766                          BLO     10$                     ;BR TO ECHO BAD INPUT
013070  126127  005212  000067          CMPB    BUFF0(R1),#'7           ;SEE IF CHAR > 7
013076  101362                          BHI     10$                     ;BR TO ECHO BAD INPUT
013100  005201                          INC     R1                      ;INCR CHAR POINTER
013102  020100                          CMP     R1,R0                   ;SEE IF MORE CHARS TO CHECK
013104  001406                          BEQ     16$                     ;BR IF ALL CHARS CHECKED
013106  126127  005212  000054          CMPB    BUFF0(R1),#',           ;SEE IF THIS IS A COMMA
013114  001353                          BNE     10$                     ;BR IF NOT A COMMA
013116  005201                          INC     R1                      ;INCR CHAR POINTER
013120  000757                          BR      14$                     ;BR TO CONTINUE CHECKING CHARS
                                        ;PUT DRIVES FOR THIS SUBSYSTEM INTO SCRATCH LIST
013122  012701  005556          16$:    MOV     #SCRLST,R1              ;GET SCRATCH DRIVE LIST ADRS
013126  005021                          CLR     (R1)+                   ;CLEAR OUT SCRATCH DRIVE LIST
013130  005021                          CLR     (R1)+
013132  005021                          CLR     (R1)+
013134  005011                          CLR     (R1)
013136  005000                          CLR     R0                      ;INIT CHAR POINTER
013140  116001  005212          18$:    MOVB    BUFF0(R0),R1            ;GET A DRIVE NO.
013144  042701  000060                  BIC     #'0,R1                  ;STRIP ASCII BITS
013150  116061  005212  005556          MOVB    BUFF0(R0),SCRLST(R1)    ;SET ASCII DRV NO.
013156  062700  000002                  ADD     #2,R0                   ;POINT TO NEXT DRIVE NO.
013162  105760  005211                  TSTB    BUFF0-1(R0)             ;SEE IF NULL CHAR YET
013166  001364                          BNE     18$                     ;BR IF NOT DONE YET
                                        ;LOAD DRIVES FROM SCRATCH LIST INTO TOTAL DRIVE LIST
013170  005000          ALLSY3: CLR     R0                      ;INIT SCRATCH DRV LIST POINTER
013172  116014  005556  ALLSY2: MOVB    SCRLST(R0),(R4)         ;MOVE THIS DRIVE INTO ACTUAL LIST
013176  001411                  BEQ     22$                     ;BR IF DRIVE NOT MARKED
```

```
3664  013200  005204                          INC     R4              ;INCR DRIVE LIST POINTER
3665  013202  110324                          MOVB    R3,(R4)+        ;PUT SUBSYS LETTER INTO HI BYTE
3666  013204  020427  005554                  CMP     R4,#DRVLST+34.  ;SEE IF DRIVE LIST OVERFLOW
3667  013210  103404                          BLO     22$             ;BR IF NO OVERFLOW YET
3668  013212  104401  010215                  TYPE    ,DROVFL         ;TYPE 'MORE THAN 16 DRIVES ENTERED !''
3669  013216  000137  012752                  JMP     ALLSYS          ;GO START OVER, ASKING FOR DRIVES
3670  013222  005200                  22$:    INC     P0              ;INCR SCRATCH DRIVE LIST POINTER
3671  013224  020027  000010                  CMP     R0,#8.          ;SEE IF DONE WITH THIS SUBSYS
3672  013230  103760                          BLO     ALLSY2          ;BR IF NOT YET
3673                                  ;ECHO DRIVES TO TEST ON THIS SUBSYSTEM
3674  013232  104401  010100                  TYPE    ,WILTST         ;TYPE 'WILL TEST DRIVE(S) ''
3675  013236  005037  005444                  CLR     SCRACH          ;INIT CHAR OUTPUT BUF
3676  013242  005000                          CLR     R0              ;INIT SCRATCH LIST POINTER
3677  013244  005001                          CLR     R1              ;INIT COUNT OF LISTED DRIVES FOR SUBSYS
3678  013246  116037  005556  005444  23$:    MOVB    SCRLST(R0),SCRACH ;SEE IF THIS DRIVE LISTED
3679  013254  001407                          BEQ     26$             ;BR IF DRIVE NOT LISTED
3680  013256  005701                          TST     R1              ;SEE IF FIRST TIME HERE
3681  013260  001402                          BEQ     24$             ;BR IF FIRST TIME
3682  013262  104401  012111                  TYPE    ,COMMA          ;TYPE A COMMA
3683  013266  104401  005444          24$:    TYPE    ,SCRACH         ;TYPE A DRIVE NO.
3684  013272  005201                          INC     R1              ;INCR. COUNT OF LISTED DRIVES FOR THIS SUBSYS
3685  013274  005200                  26$:    INC     R0              ;INCR LIST POINTER
3686  013276  020027  000010                  CMP     R0,#8.          ;SEE IF DONE CHECKING LIST YET
3687  013302  002761                          BLT     23$             ;BR IF NOT DONE YET
3688  013304  110337  010137                  MOVB    R3,ONSUBS+11.   ;SET SUBSYS NO. IN MSG
3689  013310  104401  010124                  TYPE    ,ONSUBS         ;TYPE 'ON SUBSYSTEM ''
3690  013314  005203                          INC     R3              ;INCR SUBSYSTEM NAME
3691  013316  020327  000120                  CMP     R3,#'P          ;SEE IF NAME > P
3692  013322  101035                          BHI     ASKSYS          ;BR IF YES
3693  013324  105737  003126                  TSTB    MDFLAG          ;SEE IF 200 START
3694  013330  001002                          BNE     28$             ;BR IF  NOT
3695  013332  000137  013704                  JMP     DRVTST          ;START PASS 1
3696                                  ;ASK IF THERE IS ANOTHER SUBSYSTEM TO SPECIFY
3697  013336  104401  010144          28$:    TYPE    ,ANOTHR         ;ASK FOR ANOTHER SUBSYS
3698  013342  004737  023402                  JSR     PC,RDCHRS       ;READ RESPONSE
3699  013346  012752                          ALLSYS                  ;(^C) RETURN ADDRESS
3700  013350  012752                          ALLSYS                  ;(^Z) RETURN ADDRESS
3701  013352  013336                          28$                     ;(^U) RETURN ADDRESS
3702  013354  005700                          TST     R0              ;SEE IF NULL INPUT
3703  013356  001417                          BEQ     ASKSYS          ;BR IF NULL INPUT, TO ASSUME <N>
3704  013360  022737  000131  005212          CMP     #'Y,BUFF0       ;SEE IF <Y> TYPED
3705  013366  001002                          BNE     30$             ;BR IF NOT <Y>
3706  013370  000137  013010                  JMP     ALLSY1          ;GO ASK FOR DRIVES ON NEXT SUBSYS
3707  013374  022737  000116  005212  30$:    CMP     #'N,BUFF0       ;SEE IF <N> TYPED
3708  013402  001405                          BEQ     ASKSYS          ;BR IF <N>
3709  013404  104401  005212                  TYPE    ,BUFF0          ;ECHO BAD INPUT
3710  013410  104401  001324                  TYPE    ,$QUES          ;TYPE <?> AND <CR>,<LF>
3711  013414  000750                          BR      28$             ;GO ASK AGAIN
3712
3713                                  ;204 OR 220 START - GET NAME OF SUBSYSTEM TO TEST NEXT
3714  013416  105037  001102          ASKSYS: CLRB    $TSTNM          ;SET TEST NO. = 0
3715  013422  104401  007722                  TYPE    ,TYPSYS         ;ASK FOR NEXT SUBSYS TO TEST
3716  013426  004737  023402                  JSR     PC,RDCHRS       ;READ RESPONSE
3717  013432  012752                          ALLSYS                  ;(^C) RETURN ADDRESS
3718  013434  013416                          ASKSYS                  ;(^Z) RETURN ADDRESS
3719  013436  013416                          ASKSYS                  ;(^U) RETURN ADDRESS
```

```
3720  013440  005700                          TST     R0              ;SEE IF NULL INPUT
3721  013442  001765                          BEQ     ASKSYS          ;BR IF NULL, TO ASK AGAIN
3722  013444  023727  005212  000101          CMP     BUFF0,#'A       ;SEE IF A CHAR < A TYPED
3723  013452  103005                          BHIS    6$              ;BR IF NOT
3724  013454  104401  005212          4$:     TYPE    ,BUFF0          ;ECHO BAD INPUT
3725  013460  104401  001324                  TYPE    ,$QUES          ;TYPE <?> AND <CR>,<LF>
3726  013464  000754                          BR      ASKSYS          ;GO ASK AGAIN
3727  013466  023727  005212  000120  6$:     CMP     BUFF0,#'P       ;SEE IF A CHAR > P TYPED
3728  013474  101367                          BHI     4$              ;BR IF YES
3729  013476  113737  005212  005510          MOVB    BUFF0,SUBSYS    ;STORE CURRENT SUBSYSTEM NAME
3730  013504  012700  005513                  MOV     #DRVLST+1,R0    ;SET DRIVE LIST POINTER
3731  013510  122037  005510          8$:     CMPB    (R0)+,SUBSYS    ;SEE IF A DRIVE LISTED FOR THIS SUBSYS
3732  013514  001410                          BEQ     9$              ;BR IF YES
3733  013516  005200                          INC     R0              ;INCR POINTER
3734  013520  020027  005553                  CMP     R0,#DRVLST+33.  ;SEE IF WHOLE LIST CHECKED YET
3735  013524  103771                          BLO     8$              ;BR IF NOT YET
3736  013526  104401  010622                  TYPE    ,NODRLS         ;TYPE 'NO DRIVES LISTED FOR SUBSYS''
3737  013532  000137  013416                  JMP     ASKSYS          ;GO  ASK FOR  A SUBSYS AGAIN
3738  013536  162700  000002          9$:     SUB     #2,R0           ;POINT TO LOW BYTE OF LIST ENTRY
3739  013542  010037  005554                  MOV     R0,DRVPTR       ;STORE POINTER
3740                                  ;OPEN RK06 UNIBUS ADDRESS FOR MODIFICATION
3741  013546  013746  003046                  MOV     RKBAS,-(SP)     ;PUT OLD VALUE ON STACK
3742  013552  104401  007630                  TYPE    ,RKBADR         ;TYPE 'RK06 BUS ADR = ''
3743  013556  004737  020776                  JSR     PC,GETPRM       ;TYPE OLD, GET NEW RKBAS VALUE
3744  013562  012637  003046                  MOV     (SP)+,RKBAS     ;STORE NEW VALUE
3745                                  ;OPEN RK06 VECTOR ADDRESS FOR MODIFICATION
3746  013566  013746  003050                  MOV     RKVEC,-(SP)     ;PUT OLD VALUE ON STACK
3747  013572  104401  007654                  TYPE    ,RKVADR         ;TYPE 'RK06 VEC ADR = ''
3748  013576  004737  020776                  JSR     PC,GETPRM       ;TYPE OLD, GET NEW RKVEC VALUE
3749  013602  012637  003050                  MOV     (SP)+,RKVEC     ;STORE NEW VALUE
3750                                  ;OPEN RK06 INTERRUPT HANDLER PRIORITY LEVEL FOR MODIFICATION
3751  013606  013746  003052          11$:    MOV     RKPRI,-(SP)     ;GET OLD VALUE OF PRIORITY
3752  013612  006316                          ASL     (SP)            ;GET IT INTO BITS 0-2
3753  013614  006316                          ASL     (SP)
3754  013616  006316                          ASL     (SP)
3755  013620  000316                          SWAB    (SP)
3756  013622  104401  007677                  TYPE    ,RKPRTY         ;TYPE   'RK06 PRIORITY = ''
3757  013626  004737  020776                  JSR     PC,GETPRM       ;TYPE OLD, GET NEW RKPRI VALUE
3758  013632  012600                          MOV     (SP)+,R0
3759  013634  020027  000004                  CMP     R0,#4           ;SEE IF AT LEAST LEVEL 4
3760  013640  002414                          BLT     12$             ;BR IF NEW VALUE TOO SMALL
3761  013642  020027  000007                  CMP     R0,#7           ;SEE IF LEVEL 7 OR LESS
3762  013646  003011                          BGT     12$             ;BR IF NEW VALUE TOO LARGE
3763  013650  000300                          SWAB    R0              ;GET PRIORITY INTO BITS 5-7
3764  013652  006200                          ASR     R0
3765  013654  006200                          ASR     R0
3766  013656  006200                          ASR     R0
3767  013660  010037  003052                  MOV     R0,RKPRI        ;STORE NEW VALUE
3768  013664  104401  001325                  TYPE    ,$CRLF
3769  013670  000405                          BR      DRVTST
3770  013672  104401  005212          12$:    TYPE    ,BUFF0          ;ECHO BAD INPUT
3771  013676  104401  001324                  TYPE    ,$QUES
3772  013702  000741                          BR      11$             ;GO ASK AGAIN
3773
3774  013704  113737  003155  010301  DRVTST: MOVB    PASSNO,SRPASS+18. ;SET PASS NO. FOR PRINTOUT
3775  013712  104401  010257                  TYPE    ,SRPASS         ;TYPE ''STARTING PASS X''
```

```
 3776  013716  122737  000061  003155          CMPB    #'1,PASSNO      ;SEE IF PASS 1
 3777  013724  001002                          BNE     2$              ;BR IF NOT PASS 1
 3778  013726  000137  014316                  JMP     TST1            ;START PASS 1
 3779  013732  112737  000004  001102  2$:     MOVB    #4,$TSTNM       ;SET TEST NUMBER = 4
 3780  013740  000137  015116                  JMP     TST5            ;START PASS 2
 3781
 3782
 3783                                  ;ADR 200 START - COMPILE SCRATCH LIST OF DRIVES TO SIZE
 3784  013744  005000                  DFLTDR: CLR     R0              ;CLEAR DRIVE NUMBER
 3785  013746  013703  001406                  MOV     $DEVM,R3        ;GET APT DEVICE MAP
 3786  013752  132737  000200  001351          BITB    #BIT7,$ENVM     ;SEE IF NO SIZING
 3787  013760  001002                          BNE     1$              ;BR IF NO SIZING
 3788  013762  012703  000377                  MOV     #377,R3         ;SET UP FOR SIZING
 3789  013766  012701  000001          1$:     MOV     #BIT0,R1        ;SET BIT POINTER
 3790  013772  105060  005556          2$:     CLRB    SCRLST(R0)      ;INITIALIZE DRIVE ENTRY
 3791  013776  030103                          BIT     R1,R3           ;SEE IF THIS DRIVE IS REQUESTED
 3792  014000  001405                          BEQ     4$              ;BR IF NOT
 3793  014002  110060  005556                  MOVB    R0,SCRLST(R0)   ;MARK DRIVE IN SCRATCH LIST
 3794  014006  152760  000060  005556          BISB    #'0,SCRLST(R0)  ;CONVERT NO. TO ASCII
 3795  014014  006301                  4$:     ASL     R1              ;SHIFT BIT POINTER
 3796  014016  005200                          INC     R0              ;INCREMENT DRIVE NUMBER
 3797  014020  022700  000010                  CMP     #10,R0          ;SEE IF DONE MARKING ALL DRIVES
 3798  014024  001362                          BNE     2$              ;BR IF NOT DONE YET
 3799  014026  132737  000200  001351          BITB    #BIT7,$ENVM     ;SEE IF PROGRAM SHOULD SIZE
 3800  014034  001416                          BEQ     8$              ;BR IF YES
 3801  014036  012704  005512          10$:    MOV     #DRVLST,R4      ;GET DRIVE LIST ADDRESS
 3802  014042  113703  005510                  MOVB    SUBSYS,R3       ;GET SUBSYS NAME IN R3
 3803  014046  010401                          MOV     R4,R1           ;GET DRIVE LIST ADRS
 3804  014050  010437  005554                  MOV     R4,DRVPTR       ;SET POINTER TO FIRST DRIVE TO TEST
 3805  014054  012700  000021                  MOV     #17.,R0         ;SET COUNTER
 3806  014060  005021                  11$:    CLR     (R1)+           ;CLEAR A DRIVE LIST WORD
 3807  014062  005300                          DEC     R0              ;DECREMENT COUNTER
 3808  014064  001375                          BNE     11$             ;BR IF NOT DONE YET
 3809  014066  000137  013170                  JMP     ALLSY3          ;GO MOVE SCRATCH LIST INTO LIST
 3810                                  ;REMOVE NON-EXISTENT DRIVES FROM SCRATCH LIST
 3811  014072  005000                  8$:     CLR     R0              ;CLEAR DRIVE NUMBER
 3812  014074  105760  005556          12$:    TSTB    SCRLST(R0)      ;SEE IF THIS DRIVE IS MARKED IN SCRATCH LIST
 3813  014100  001475                          BEQ     14$             ;BR IF NOT MARKED
 3814  014102  010037  005430                  MOV     R0,DRIVE        ;SET DRIVE NUMBER FOR SCNDRV
 3815  014106  004737  021140                  JSR     PC,INITSS       ;CLEAR DRIVER AND INIT S.S.
 3816  014112  042712  000100          3$:     BIC     #IE,(R2)        ;DISABLE RK06 INTERRUPT
 3817  014116  113737  005430  000010          MOVB    DRIVE,RKCS2
 3818  014124  105737  003133                  TSTB    TYPFMT          ;SEE IF RK07
 3819  014130  001003                          BNE     5$              ;BR IF YES
 3820  014132  012712  000001                  MOV     #GO,(R2)        ;ELSE SET GO BIT FOR RK06 IN RKCS1
 3821  014136  000402                          BR      7$
 3822  014140  012712  002001          5$:     MOV     #<CDT!GO>,(R2)  ;ELSE SET GO FOR RK07 IN RKCS1
 3823  014144  005037  005444          7$:     CLR     SCRACH          ;CLEAR STALL CTR
 3824  014150  005237  005444          15$:    INC     SCRACH          ;INCREMENT STALL COUNTER
 3825  014154  001375                          BNE     15$             ;STALL FOR SEVERAL MILLI-SEC
 3826  014156  032762  001000  000010          BIT     #MDS,RKCS2(R2)  ;SEE IF MDS ERROR
 3827  014164  001417                          BEQ     18$             ;BR IF NOT
 3828  014166  112765  000101  000001          MOVB    #SELDRV,P.CMND(R5) ;SET CMND FOR ERROR REPORT
 3829  014174  011265  000016                  MOV     (R2),P.CS1(R5)  ;GET RKCS1 FOR REPORT
 3830  014200  004737  037632                  JSR     PC,I.CST1       ;GET OTHER RK611 REGS FOR REPORT
 3831  014204  004737  031212                  JSR     PC,REPSUP       ;SET UP ERROR REPORT
```

```
3832  014210  104052                          ERROR   52              ;REPORT MDS ERROR
3833  014212  052737  000200  005424          BIS     #ABORT,RECODE   ;SET ABORT FLAG
3834  014220  000137  033324                  JMP     ALLTRM          ;GO ABORT TESTING
3835  014224  004737  021140          18$:    JSR     PC,INITSS       ;INIT THE S.S.
3836  014230  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;SET READ STATUS COMMAND
3837  014236  012737  020450  003056          MOV     #NEDHDL,A.ABNL  ;SET NED ABNORMAL RETURN ADR
3838  014244  012737  000377  005476          MOV     #377,NEWON      ;INIT ONLINE INDICATOR
3839  014252  004737  030124                  JSR     PC,DRVCAL       ;READ DRIVE STATUS
3840  014256  012737  031472  003056          MOV     #ERRHDL,A.ABNL  ;RESTORE ABNORMAL RETURN ADDRESS
3841  014264  022737  000377  005476          CMP     #377,NEWON      ;SEE IF THIS DRIVE EXISTS
3842  014272  001006                          BNE     16$             ;BR IF NO
3843  014274  005200                  14$:    INC     R0              ;RETURN HERE IF DRIVE IS USEABLE
3844  014276  022700  000010                  CMP     #10,R0          ;SEE IF DONE CHECKING LIST
3845  014302  001274                          BNE     12$             ;BR IF NOT DONE YET
3846  014304  000137  014036                  JMP     10$             ;GO BACK TO LIST SELECTED DRIVES
3847  014310  105060  005556          16$:    CLRB    SCRLST(R0)      ;REMOVE INVALID DRIVE FROM LIST
3848  014314  000767                          BR      14$             ;CONTINUE CHECKING THE LIST
3849
3850
3851
3852
3853
3854                                  ;;*****************************************************************
3855                                  ;*TEST 1        MOUNTING OF TEST CARTRIDGE FOR PASS 1
3856                                  ;*
3857                                  ;*      THE PROGRAM TYPES "STARTING PASS 1". THEN, THE OPERATOR
3858                                  ;*      MOUNTS THE PACK ON THIS DRIVE AND MANUALLY LOADS THE HEADS,
3859                                  ;*      AS DIRECTED BY THE PROGRAM. IT THEN CHECKS THIS DRIVE
3860                                  ;*      FOR VALID STATUS, AND READS THE HEADER ON SECTOR 0, TO
3861                                  ;*      DETERMINE THE FORMAT. THE FORMAT IS STORED IN "FORMAT".
3862                                  ;*      ALSO, THE PROGRAM READS AND STORES THE BAD SECTOR FILES, AND
3863                                  ;*      IT TYPES THE DRIVE AND CARTRIDGE SERIAL NUMBERS.
3864                                  ;*
3865                                  ;;*****************************************************************
3866  014316  000004                  TST1:   SCOPE
3867  014320  012737  000001  001334          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
3868  014326  004737  022020                  JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
3869  014332  004737  021140                  JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
3870  014336  004737  020744                  JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
3871  014342  004737  022066                  JSR     PC,MTCART
3872
3873
3874
3875
3876
3877                                  ;;*****************************************************************
3878                                  ;*TEST 2        BASIC READ/WRITE DATA TEST
3879                                  ;*
3880                                  ;*      THE PROGRAM PERFORMS A WRITE AND WRITE CHECK OPERATION
3881                                  ;*      USING A "WORST CASE" DATA PATTERN, AT THE APPROPRIATE SECTOR
3882                                  ;*      FOR THIS DRIVE (FROM TABLE A) ON ALL SURFACES. THE PURPOSE
3883                                  ;*      OF THIS OPERATION IS TO VERIFY THE BASIC READ/WRITE
3884                                  ;*      CAPABILITY OF THE DRIVE ON PASS 1. THE ENTIRE SECTOR
3885                                  ;*      IS WRITTEN WITH THE REPETITION OF THE DATA PATTERN SHOWN
3886                                  ;*      IN TABLE B.
3887                                  ;*
```

```
                                        ;;*****************************************************************
014346  000004                        TST2:   SCOPE
014350  012737  000002  001334                MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
014356  004737  022020                        JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
014362  004737  021140                        JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
014366  004737  020744                        JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
014372  112737  000001  003127                MOVB    #1,TSTING               ;ALLOW TTY ESCAPE
                                        ;LOAD RWBUF WITH DATA PATTERN
014400  004737  024460                        JSR     PC,LODSEC
                                        ;SET UP PARAMETERS
014404  113700  003154                        MOVB    LOGDRV,R0       ;GET LOGICAL DRIVE NUMBER
014410  006300                        ASL     R0              ;CONVERT IT TO INDEX
014412  006300                        ASL     R0
014414  105737  003133                        TSTB    TYPFMT          ;SEE IF RK07
014420  001011                        BNE     1$              ;BR IF YES
014422  016065  005566  000002                MOV     TABLEA(R0),P.CYLN(R5) ;GET CYL # FOR THIS DRIVE
014430  062700  000002                        ADD     #2,R0           ;INCREMENT INDEX
014434  116065  005566  000004                MOVB    TABLEA(R0),P.SECT(R5) ;GET SECTOR # FOR THIS DRIVE
014442  000410                        BR      2$
                                
014444  016065  005666  000002  1$:     MOV     TABL7A(R0),P.CYLN(R5)   ;GET RK07 TABLE
014452  062700  000002                        ADD     #2,R0
014456  116065  005666  000004                MOVB    TABL7A(R0),P.SECT(R5)   ;GET SECTOR # FOR THE DRV
                                
014464  012765  053330  000010  2$:     MOV     #RWBUF,P.BALO(R5)       ;SET BUS ADDRESS
014472  012765  177400  000012                MOV     #-400,P.WC(R5)  ;SET WORD COUNT FOR 1 SECTOR
                                        ;WRITE THE DATA
014500  112765  000123  000001  6$:     MOVB    #WRDATA,P.CMND(R5)      ;SET WRITE COMMAND
014506  004737  030124                        JSR     PC,DRVCAL       ;WRITE THE SECTOR
                                        ;PERFORM WRITE CHECK OF SECTOR
014512  112765  000131  000001                MOVB    #WRTCHK,P.CMND(R5)      ;SET WRITE CHECK COMMAND
014520  004737  030124                        JSR     PC,DRVCAL       ;PERFORM THE WRITE CHECK
                                        ;INCREMENT TRACK ADDRESS
014524  105265  000005                        INCB    P.TRCK(R5)      ;INCREMENT TRACK NO.
014530  126527  000005  000002                CMPB    P.TRCK(R5),#2   ;SEE IF ALL SURFACES WRITTEN YET
014536  101760                        BLOS    6$              ;BR IF NOT YET
                                
                                
                                
                                
                                        ;;*****************************************************************
                                        ;*TEST 3        WRITE OVRWRT AND COMPAT CYL BLOCKS FOR CURRENT DRIVE
                                        ;*      ALL SECTORS ARE WRITTEN FOR THE CURRENT DRIVE WITHIN THE
                                        ;*      CYLINDER BLOCKS IN TABLEC ON ALL SURFACES, USING A SINGLE
                                        ;*      REPEATED WORD OF THE PSEUDO-RANDOM DATA PATTERN IN TABLEG. LOGICAL DRIVE 0
                                        ;*      USES WORD 0, LOGICAL DRIVE 14 USES WORD 14, ETC.  THESE CYLINDER
                                        ;*      BLOCKS ARE THE OVERWRITE AND COMPATIBILITY TEST DATA TO
                                        ;*      BE USED IN PASS 2.
                                        ;;*****************************************************************
014540  000004                        TST3:   SCOPE
014542  012737  000003  001334                MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
014550  004737  022020                        JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
014554  004737  021140                        JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
014560  004737  020744                        JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
014564  004737  022620                        JSR     PC,INTBLD       ;LOAD INITIAL VALUES INTO TABLED
014570  005000                        CLR     R0              ;INITIALIZE TABLEC INDEX TO 0
```

```
3944  014572  105737  003133          4$:     TSTB    TYPFMT          ;SEE IF RK07
3945  014576  001013                          BNE     1$              ;BR IF YES
3946  014600  016065  006026  000002          MOV     TABLEC(R0),P.CYLN(R5) ;GET START OVRWRT CYL FROM TABLEC
3947  014606  004737  023064                  JSR     PC,WRTBLK       ;WRITE THIS OVRWRT CYL BLK ON ALL SURF'S
3948  014612  062700  000002                  ADD     #2,R0           ;INCREMENT TABLEC POINTER
3949  014616  016065  006026  000002          MOV     TABLEC(R0),P.CYLN(R5) ;GET START COMPAT CYL FROM TABLEC
3950  014624  000412                          BR      2$
3951
3952  014626  016065  006060  000002  1$:     MOV     TABL7C(R0),P.CYLN(R5)
3953  014634  004737  023064                  JSR     PC,WRTBLK
3954  014640  062700  000002                  ADD     #2,R0
3955  014644  016065  006060  000002          MOV     TABL7C(R0),P.CYLN(R5)
3956  014652  004737  023064          2$:     JSR     PC,WRTBLK       ;WRITE THIS COMPAT CYL BLK ON ALL SURF'S
3957  014656  062700  000002                  ADD     #2,R0           ;INCREMENT TABLEC POINTER
3958  014662  020027  000030                  CMP     R0,#30          ;SEE IF ON LAST CURRENT ZONE YET
3959  014666  002003                          BGE     8$              ;BR IF YES
3960  014670  004737  023000                  JSR     PC,ROTBLD       ;ROTATE TABLED SEVERAL SECTORS
3961  014674  000736                          BR      4$              ;CONTINUE WRITING BLOCKS
3962  014676  004737  022620          8$:     JSR     PC,INTBLD       ;RE-LOAD INITIAL VALUES INTO TABLED
3963  014702  105737  003133                  TSTB    TYPFMT          ;SEE IF RK07
3964  014706  001004                          BNE     3$              ;BR IF YES
3965  014710  016065  006026  000002          MOV     TABLEC(R0),P.CYLN(R5) ;GET START OVRWRT CYL FROM TABLEC
3966  014716  000403                          BR      5$
3967
3968  014720  016065  006060  000002  3$:     MOV     TABL7C(R0),P.CYLN(R5)
3969  014726  004737  023064          5$:     JSR     PC,WRTBLK       ;WRITE THIS OVRWRT CYL BLK ON ALL SURF'S
3970
3971
3972
3973                                  ;;*****************************************************************
3974                                  ;*TEST 4        DISMOUNTING OF TEST CARTRIDGE IN PASS 1
3975                                  ;*
3976                                  ;*      THE PROGRAM DIRECTS THE OPERATOR TO UNLOAD THE DRIVE CURRENTLY
3977                                  ;*      UNDER TEST, AND TO REMOVE THE TEST PACK. THEN, THE PROGRAM
3978                                  ;*      DOES ONE OF 4 THINGS :
3979                                  ;*      (1) IF THERE IS ANOTHER DRIVE TO TEST ON THIS SUBSYSTEM, THE
3980                                  ;*      PROGRAM JUMPS TO TEST 1 FOR THE NEXT DRIVE.
3981                                  ;*      (2) IF THERE IS ONLY ONE SUBSYSTEM, (FROM 200 START), AND
3982                                  ;*      IF PASS 1 HAS BEEN COMPLETED ON ALL DRIVES, THE PROGRAM STARTS
3983                                  ;*      PASS 2 ON THE FIRST DRIVE, BY JUMPING TO TEST 5.
3984                                  ;*      (3) IF THERE IS ANOTHER SUBSYSTEM TO PERFORM PASS 1 ON, THE PROGRAM
3985                                  ;*      TELLS THE OPERATOR TO RESTART AT ADDRESS 204, AND THEN IT HALTS.
3986                                  ;*      (4) IF THERE ARE NO MORE DRIVES TO TEST IN PASS 1 ON ANY
3987                                  ;*      OTHER SUBSYSTEM, THE PROGRAM TELLS THE OPERATOR TO
3988                                  ;*      RESTART AT ADDRESS 220 FOR PASS 2 ON THE NEXT SUBSYSTEM, AND
3989                                  ;*      THEN IT HALTS.
3990                                  ;*
3991                                  ;*****************************************************************
3992  014732  000004                  TST4:   SCOPE
3993  014734  012737  000004  001334          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
3994  014742  004737  022020                  JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
3995  014746  004737  021140                  JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
3996  014752  004737  020744                  JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
3997  014756  105037  003127                  CLRB    TSTING          ;DON'T ALLOW TTY ESCAPE
3998  014762  004737  022450                  JSR     PC,DMCART       ;DIRECT OPERATOR TO DISMOUNT PACK
3999  014766  062737  000002  005554          ADD     #2,DRVPTR       ;INCREMENT DRIVE LIST POINTER
```

```
 4000  014774  013700  005554                   MOV     DRVPTR,R0
 4001  015000  005710                           TST     (R0)              ;SEE IF ALL DONE WITH PASS 1 YET
 4002  015002  001420                           BEQ     8$                ;BR IF DONE WITH PASS 1
 4003  015004  126037  000001  005510           CMPB    1(R0),SUBSYS      ;SEE IF DONE WITH THIS SUBSYS YET
 4004  015012  001004                           BNE     4$                ;BR IF DONE WITH THIS SUBSYS
 4005  015014  105037  001102                   CLRB    $TSTNM            ;CLEAR TEST NUMBER
 4006  015020  000137  014316                   JMP     TST1              ;DO PASS 1 ON NEXT DRIVE OF SUBSYS
 4007  015024  116037  000001  010467  4$:      MOVB    1(R0),STR204+39.  ;PUT SUBSYS NAME INTO MSG
 4008  015032  104401  010420          5$:      TYPE    ,STR204           ;TYPE RESTART MSG FOR PASS 1 ON NEXT SUBSYS
 4009  015036  000000                  6$:      HALT                      ;HALT THE PROCESSOR SO THAT
 4010  015040  000137  000204                   JMP     204               ;  OPERATOR CAN RESTART AT PROPER ADRS
 4011  015044  012737  005512  005554  8$:      MOV     #DRVLST,DRVPTR    ;RE-INIT DRIVE LIST POINTER
 4012  015052  105737  003126                   TSTB    MDFLAG            ;SEE IF 200 START
 4013  015056  001007                           BNE     12$               ;BR IF NOT
 4014  015060  112737  000062  010301           MOVB    #'2,SRPASS+18.    ;SET PASS NO. = 2 FOR PRINTOUT
 4015  015066  104401  010257                   TYPE    ,SRPASS           ;TYPE "STARTING PASS 2"
 4016  015072  000137  015116                   JMP     TST5              ;GO START PASS 2
 4017  015076  112737  000101  010544  12$:     MOVB    #'A,STR220+39.    ;PUT SUBSYS NAME INTO MSG
 4018  015104  104401  010475                   TYPE    ,STR220           ;TYPE RESTART MSG FOR PASS 2 ON FIRST SUBSYS
 4019  015110  000000                           HALT                      ;HALT PROCESSOR SO THAT OPERATOR CAN RESTART
 4020  015112  000137  000220                   JMP     220               ;   AT THE PROPER ADRS
 4021
 4022
 4023
 4024                                   ;;*****************************************************************
 4025                                   ;*TEST 5        MOUNTING OF TEST CARTRIDGE FOR PASS 2
 4026                                   ;*
 4027                                   ;*      THE PROGRAM TYPES "STARTING PASS 2". THEN, THE OPERATOR
 4028                                   ;*      MOUNTS THE PACK ON THIS DRIVE AND MANUALLY LOADS THE HEADS,
 4029                                   ;*      AS DIRECTED BY THE PROGRAM. IT THEN CHECKS THIS DRIVE
 4030                                   ;*      FOR VALID STATUS, AND READS THE HEADER ON SECTOR 0, TO
 4031                                   ;*      DETERMINE THE FORMAT. THE FORMAT IS STORED IN "FORMAT".
 4032                                   ;*      ALSO, THE PROGRAM READS AND STORES THE BAD SECTOR FILES, AND
 4033                                   ;*      IT TYPES THE DRIVE AND CARTRIDGE SERIAL NUMBERS (IF STARTED
 4034                                   ;*      AT ADDRESS 204 OR 220).
 4035                                   ;*
 4036                                   ;;*****************************************************************
 4037  015116  000004                   TST5:   SCOPE
 4038  015120  012737  000005  001334           MOV     #5,$TESTN         ;;SET TEST NUMBER IN APT MAIL BOX
 4039  015126  004737  022020                   JSR     PC,SETTUP         ;SET UP FOR LOOP ON ERROR
 4040  015132  004737  021140                   JSR     PC,INITSS         ;INITIALIZE DRIVER PARAMS AND SUB-SYS
 4041  015136  004737  020744                   JSR     PC,PREPKB         ;PREPARE FOR POSSIBLE KBD INPUT
 4042  015142  112737  000062  003155           MOVB    #'2,PASSNO        ;SET PASS NO. = 2
 4043  015150  004737  022066                   JSR     PC,MTCART
 4044
 4045
 4046
 4047                                   ;;*****************************************************************
 4048                                   ;*TEST 6        OVERWRITE TEST
 4049                                   ;*
 4050                                   ;*      THE PROGRAM PROCEEDS TO TEST THIS DRIVE'S OVERWRITE CAPABILITY.
 4051                                   ;*      FIRST, THE APPROPRIATE CYLINDERS IN TABLE E FOR THIS DRIVE
 4052                                   ;*      ARE OVERWRITTEN, ON EACH SURFACE. THE DATA USED IS A
 4053                                   ;*      REPETITION OF A SINGLE WORD OF THE PATTERN IN TABLE G,
 4054                                   ;*      DRIVE 0 USES WORD 0, DRIVE 1 USES WORD 1, DRIVE 10 USES
 4055                                   ;*      WORD 10, ETC.
```

```
                                        ;*      THEN, EACH CYLINDER OVERWRITTEN IS READ BACK BY THIS DRIVE
                                        ;*      WITH THE FOLLOWING RANGE OF OFFSET VALUES :  100, 200, 300,
                                        ;*      400,500,600,700,800,900,1000,1100,1200 MICRO-IN., IN BOTH
                                        ;*      THE (+) AND (-) DIRECTIONS.
                                        ;*      THE PROGRAM SCANS FOR DATA CHECK ERRORS DURING THIS READ, AND IF
                                        ;*      ONE OCCURS, THE PROGRAM DETERMINES WHICH DRIVE'S DATA HAD NOT
                                        ;*      BEEN CORRECTLY OVERWRITTEN, AND A SCORE FOR THAT DRIVE IS
                                        ;*      DECREMENTED. THEN, THE TRANSFER IS CONTINUED AT THE NEXT SECTOR,
                                        ;*      WITH THAT OFFSET VALUE. THE READS ARE DONE WITH ALL OF THE ABOVE
                                        ;*      OFFSETS APPLIED, AND A SEPARATE SCORE FOR EACH DRIVE IS KEPT,
                                        ;*      WHILE THE CURRENT DRIVE IS PERFORMING THE OVERWRITES.
                                        ;*      FOR EACH TRACK (0,1,2), SCORES ARE AVERAGED FOR EACH OFFSET
                                        ;*       DIRECTION, OVER ALL CYLINDERS TESTED.
                                        ;*
                                        ;;**********************************.**********************************
015154  000004                  TST6:   SCOPE
015156  012737  000006  001334          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
015164  004737  022020                  JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
015170  004737  021140                  JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
015174  004737  020744                  JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
015200  112737  000001  003127          MOVB    #1,TSTING       ;ALLOW TTY ESCAPE
015206  105037  003147                  CLRB    DIF100          ;CLEAR "CYL DIF = 100/200" FLAG
                                ;OVERWRITE THE APPROPRIATE CYLS FOR THIS DRIVE
015212  112765  000117  000001          MOVB    #SEEK,P.CMND(R5) ;SET SEEK COMMAND
015220  105065  000002                  CLRB    P.CYLN(R5)      ;SET CYL = 0
015224  004737  030124                  JSR     PC,DRVCAL       ;SEEK TO CYL 0
015230  012703  000007                  MOV     #7,R3           ;SET COUNTER = 7
015234  112765  000123  000001          MOVB    #WRDATA,P.CMND(R5) ;SET WRITE DATA COMMAND
015242  113700  003154                  MOVB    LOGDRV,R0       ;GET LOGICAL DRIVE NUMBER
015246  010065  000002                  MOV     R0,P.CYLN(R5)   ;SET STARTING OVERWRITE CYL FOR THIS DRIVE
015252  006300                          ASL     R0              ;GET DATA PATTERN INDEX
015254  062700  006512                  ADD     #TABLEG,R0      ;GET PATTERN WORD ADRS
015260  011037  053330                  MOV     (R0),RWBUF      ;PUT DATA WORD INTO BUFFER
015264  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;SET BUS ADDRESS
015272  052765  100000  000014          BIS     #DTBAII,P.PRST(R5) ;SET BUS ADRS INCREMENT INHIBIT
015300  012765  137000  000012          MOV     #-16896.,P.WC(R5) ;SET 22-SECTOR WORD COUNT
015306  105737  003134                  TSTB    FORMAT          ;DETERMINE THE FORMAT
015312  001403                          BEQ     6$              ;BR IF 22 SECTORS
015314  012765  142000  000012          MOV     #-15360.,P.WC(R5) ;SET 20-SECTOR WORD COUNT
015322  004737  027456          6$:     JSR     PC,SVPRMS       ;SAVE PARAMETERS OF TRANSFER
015326  004737  027546                  JSR     PC,TRNSFR       ;WRITE THE DATA
015332  063765  024370  000002          ADD     HOLD3,P.CYLN(R5) ;INCR THE CYL NO.
015340  005303                          DEC     R3              ;SEE IF DONE WITH ALL CYLS YET
015342  001367                          BNE     6$              ;BR IF NOT YET
                                ;READ OVERWRITE CYLS WITH RANGE OF OFFSETS
015344  112737  000001  003143          MOVB    #1,TSTTYP       ;SET INDICATOR FOR OVRWRT TEST
015352  012700  006612                  MOV     #NOSCR0,R0      ;GET STARTING ADRS OF OVRWRT SCORES
015356  005020                  10$:    CLR     (R0)+           ;INIT SCORE TO 0
015360  020027  007112                  CMP     R0,#NOSCR0+192. ;SEE IF ALL SCORES CLEARED YET
015364  103774                          BLO     10$             ;BR IF NOT DONE YET
015366  012703  000007                  MOV     #7,R3           ;SET CYL CNTR = 7
015372  113700  003154                  MOVB    LOGDRV,R0       ;GET LOGICAL DRIVE NO.
015376  010065  000002                  MOV     R0,P.CYLN(R5)   ;GET START OVRWRT CYL FOR THIS DRIVE
015402  004737  022620          11$:    JSR     PC,INTBLD       ;INIT TABLED
015406  004737  017250          12$:    JSR     PC,REDOFS       ;DO READS WITH OFFSETS ON THIS CYL
015412  004737  017614                  JSR     PC,ADSCOR       ;ADD SCRATCH SCORES TO SUM
```

```
 4112  015416  063765  024370  000002          ADD     HOLD3,P.CYLN(R5) ;INCR CYL NO.
 4113  015424  112737  000001  003147          MOVB    #1,DIF100        ;SET "CYL DIF = 100/200" FLAG
 4114  015432  005303                          DEC     R3               ;DECR CYL CNTR
 4115  015434  004737  023000                  JSR     PC,ROTBLD        ;ROTATE TABLED FOR NEXT CURRENT ZONE
 4116  015440  020327  000001                  CMP     R3,#1            ;SEE IF ON LAST ZONE YET
 4117  015444  003360                          BGT     12$              ;BR IF NOT YET
 4118  015446  001755                          BEQ     11$              ;BR IF ON LAST CYL
 4119  015450  004737  017774                  JSR     PC,AVSCOR        ;COMPUTE AVERAGE OVERWRITE SCORES
 4120  015454  105037  003143                  CLRB    TSTTYP           ;CLEAR OVRWRT TEST INDICATOR
 4121
 4122
 4123
 4124                                  ;;*****************************************************************
 4125                                  ;*TEST 7        DRIVE SELF-TEST
 4126                                  ;*
 4127                                  ;*      THE PROGRAM EVALUATES THE DRIVE'S ABILITY TO WRITE AND READ
 4128                                  ;*      ITS OWN DATA, AT VARIOUS ANGULAR POSITIONS ON THE PACK. FIRST,
 4129                                  ;*      SECTORS 4,11,16, AND 23 OF THE APPROPRIATE CYLINDERS SHOWN
 4130                                  ;*      IN TABLEF FOR THIS DRIVE ARE WRITTEN WITH THE DATA PATTERN
 4131                                  ;*      SHOWN IN TABLEB, FOR ALL SURFACES. THEN, THE SECTORS ARE READ
 4132                                  ;*      WITH THE FOLLOWING OFFSETS APPLIED : 100,200,300,400,500,600
 4133                                  ;*      700,800,900,1000,1100,1200 MICRO-IN., IN (+) AND (-) DIRECTIONS.
 4134                                  ;*      THE PROGRAM SCANS FOR DATA DHECK ERRORS DURING EACH READ, AND IT
 4135                                  ;*      COMPUTES A SCORE WHICH IS PROPORTIONAL TO THE FAILING OFFSET
 4136                                  ;*      MAGNITUDE .  THEN, THE SCORES FOR ALL SECTORS READ ARE AVERAGED,
 4137                                  ;*      TO COME UP WITH A DRIVE SELF-TEST SCORE FOR EACH OFFSET
 4138                                  ;*      DIRECTION, FOR EACH SURFACE.
 4139                                  ;*      THIS SCORE IS SAVED FOR LATER USE, TO BECOME THE STANDARD
 4140                                  ;*      FOR THE COMPATIBILITY DATA READS WHICH ARE TO FOLLOW.
 4141                                  ;*
 4142                                  ;;*****************************************************************
 4143  015460  000004                  TST7:   SCOPE
 4144  015462  012737  000007  001334          MOV     #7,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
 4145  015470  004737  022020                  JSR     PC,SETTUP        ;SET UP FOR LOOP ON ERROR
 4146  015474  004737  021140                  JSR     PC,INITSS        ;INITIALIZE DRIVER PARAMS AND SUB-SYS
 4147  015500  004737  020744                  JSR     PC,PREPKB        ;PREPARE FOR POSSIBLE KBD INPUT
 4148  015504  005037  007412                  CLR     NSSCR0           ;INIT SELF-TEST SCORES TO 0
 4149  015510  005037  007414                  CLR     NSSCR1
 4150  015514  005037  007416                  CLR     NSSCR2
 4151  015520  005037  007420                  CLR     PSSCR0
 4152  015524  005037  007422                  CLR     PSSCR1
 4153  015530  005037  007424                  CLR     PSSCR2
 4154  015534  105037  003147                  CLRB    DIF100           ;CLEAR "CYL DIF = 100/200" FLAG
 4155  015540  012703  000006                  MOV     #6,R3            ;SET CYL COUNTER = 6
 4156  015544  113700  003154                  MOVB    LOGDRV,R0        ;GET LOGICAL DRIVE NO.
 4157  015550  062700  000060                  ADD     #60,R0           ;COMPUTE STARTING CYL NO.
 4158  015554  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;SET BUS ADRS
 4159  015562  004737  024460                  JSR     PC,LODSEC        ;LOAD R/W BUF WITH DATA PATTERN
 4160  015566  012765  177400  000012          MOV     #-256.,P.WC(R5)  ;SET WORD COUNT FOR 1 SECTOR
 4161  015574  112765  000004  000004  3$:     MOVB    #4,P.SECT(R5)    ;SET SECTOR = 4
 4162  015602  105065  000005          4$:     CLRB    P.TRCK(R5)       ;SET TRACK = 0
 4163  015606  105037  007432                  CLRB    NSCOR0           ;INIT SCRATCH SCORES TO 0
 4164  015612  105037  007452                  CLRB    NSCOR1
 4165  015616  105037  007472                  CLRB    NSCOR2
 4166  015622  105037  007512                  CLRB    PSCOR0
 4167  015626  105037  007532                  CLRB    PSCOR1
```

```
 4168  015632  105037  007552                  CLRB    PSCOR2
 4169  015636  105037  003143          6$:     CLRB    TSTTYP          ;CLEAR OFFSET TEST INDICATOR
 4170  015642  005065  000002                  CLR     P.CYLN(R5)      ;SET CYL = 0
 4171  015646  112765  000117  000001          MOVB    #SEEK,P.CMND(R5)        ;SET SEEK COMMAND
 4172  015654  004737  030124                  JSR     PC,DRVCAL       ;PERFORM SEEK TO 0
 4173  015660  010065  000002                  MOV     R0,P.CYLN(R5)   ;SET CURRENT CYLINDER NUMBER
 4174  015664  112765  000123  000001          MOVB    #WRDATA,P.CMND(R5) ;SET WRITE DATA COMMAND
 4175  015672  004737  030124                  JSR     PC,DRVCAL       ;WRITE A SECTOR
 4176  015676  112737  000003  003143          MOVB    #3,TSTTYP       ;SET INDICATOR FOR SELF-TEST
 4177  015704  004737  017250                  JSR     PC,REDOFS       ;READ SECTOR WITH RANGE OF OFFSETS
 4178  015710  105265  000005                  INCB    P.TRCK(R5)      ;INCREMENT TRACK
 4179  015714  126527  000005  000002          CMPB    P.TRCK(R5),#2   ;SEE IF ALL TRACKS DONE ON THIS CYL
 4180  015722  101745                          BLOS    6$              ;BR IF NOT YET
 4181  015724  004737  017614                  JSR     PC,ADSCOR       ;ADD SCRATCH SCORES TO SUMS
 4182  015730  062765  000005  000004          ADD     #5,P.SECT(R5)   ;INCREMENT SECTOR BY 5
 4183  015736  126527  000004  000023          CMPB    P.SECT(R5),#23  ;SEE IF ALL SECTORS DONE ON THIS TRACK
 4184  015744  101716                          BLOS    4$              ;BR IF NOT YET
 4185  015746  063700  024370                  ADD     HOLD3,R0        ;INCR CYL NO. BY 100/200
 4186  015752  005303                          DEC     R3              ;DECR CYL CNTR
 4187  015754  001307                          BNE     3$              ;BR IF NOT DONE WITH ALL CYLS YET
 4188  015756  004737  017774                  JSR     PC,AVSCOR       ;COMPUTE AVG SELF-TEST SCORES
 4189  015762  105037  003143                  CLRB    TSTTYP          ;CLEAR OVRWRT TEST INDICATOR
 4190
 4191
 4192
 4193                                  ;;*****************************************************************
 4194                                  ;*TEST 10       COMPATIBILITY DATA READ TEST
 4195                                  ;*
 4196                                  ;*      HAVING ESTABLISHED A SELF-TEST SCORE FOR THIS DRIVE, THE
 4197                                  ;*      PROGRAM PROCEEDS TO PERFORM THE COMPATIBILITY DATA READS
 4198                                  ;*      OF THE PATTERNS WRITTEN BY ALL THE DRIVES IN EACH CYLINDER
 4199                                  ;*      BLOCK (ON EACH SURFACE). EACH COMPATIBILITY CYLINDER BLOCK
 4200                                  ;*      SHOWN IN TABLE C IS READ  WITH THE FOLLOWING OFFSET VALUES :
 4201                                  ;*      100,200,300,400,500,600,700,800,900,1000,1100,1200 MICRO-IN.,
 4202                                  ;*      IN (+) AND (-) DIRECTIONS. THE PROGRAM SCANS FOR READ ERRORS
 4203                                  ;*      DURING EACH READ, AND IF ONE OCCURS, THE PROGRAM DETERMINES
 4204                                  ;*      WHICH DRIVE'S DATA WAS BEING READ AT THAT INSTANT AND A SCORE
 4205                                  ;*      FOR THAT DRIVE IS DECREMENTED. THEN, THE TRANSFER IS CONTINUED
 4206                                  ;*      AT THE NEXT SECTOR, WITH THAT OFFSET VALUE. THE READS ARE
 4207                                  ;*      DONE WITH ALL OF THE OFFSETS APPLIED, AND A SEPARATE SCORE
 4208                                  ;*      FOR EACH DRIVE IS KEPT, WHILE THE CURRENT DRIVE IS READING
 4209                                  ;*      THE COMPATIBILITY DATA. THEN, EACH SCORE IS APPROPRIATELY
 4210                                  ;*      ADJUSTED TO REFLECT THE SELF-TEST SCORE FOR THE CURRENT DRIVE
 4211                                  ;*      ON THAT SURFACE. THE SCORES ARE THEN AVERAGED OVER ALL CYLINDER
 4212                                  ;*      BLOCKS. EACH SCORE IS PROPORTIONAL TO THE  CAPABILITY OF THE
 4213                                  ;*      CURRENT DRIVE TO SUCCESSFULLY READ THE DATA WRITTEN BY ONE OF
 4214                                  ;*      THE OTHER DRIVES, AND SCORES ARE COMPUTED SEPARATELY FOR EACH
 4215                                  ;*      SURFACE (TRACK). THUS, THE SCORES REVEAL WHICH DRIVES ARE
 4216                                  ;*      INVOLVED IN A SITUATION IN WHICH A PARTICULAR DRIVE HAS
 4217                                  ;*      DIFFICULTY IN READING THE DATA OF ONE OR SEVERAL OTHER DRIVES.
 4218                                  ;*
 4219                                  ;;*****************************************************************
 4220  015766  000004                  TST10:  SCOPE
 4221  015770  012737  000010  001334          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 4222  015776  004737  022020                  JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
 4223  016002  004737  021140                  JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
```

```
 4224  016006  004737  020744                  JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
 4225  016012  112737  000002  003143          MOVB    #2,TSTTYP       ;SET INDIC FOR READ TEST
 4226  016020  105037  003147                  CLRB    DIF100          ;CLEAR "CYL DIF = 100/200" FLAG
 4227  016024  012700  007112                  MOV     #NCSCR0,R0      ;GET ADRS OF COMPAT READ SCORES
 4228  016030  005020                  6$:     CLR     (R0)+           ;INIT A SCORE TO 0
 4229  016032  020027  007412                  CMP     R0,#NCSCR0+192. ;SEE IF ALL SCORES CLEARED YET
 4230  016036  103774                          BLO     6$              ;BR IF NOT YET
 4231  016040  012703  000006                  MOV     #6,R3           ;SET CYL BLK CNTR = 6
 4232  016044  004737  022620                  JSR     PC,INTBLD       ;INIT TABLE D
 4233  016050  013701  024372                  MOV     HOLD4,R1        ;SET START CYL = 60/160
 4234  016054  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;SET BUS ADRS
 4235  016062  052765  100000  000014          BIS     #DTBAII,P.PRST(R5) ;SET BUS ADRS INCR INHIBIT
 4236  016070  012765  137000  000012          MOV     #-16896.,P.WC(R5) ;SET 22-SECTOR WORD COUNT
 4237  016076  105737  003134                  TSTB    FORMAT          ;TEST THE FORMAT
 4238  016102  001403                          BEQ     10$             ;BR IF 22 SECTORS
 4239  016104  012765  142000  000012          MOV     #-15360.,P.WC(R5) ;SET 20-SECTOR WORD COUNT
 4240  016112  012700  000020          10$:    MOV     #16.,R0         ;INIT CYL CNTR = 16.
 4241  016116  010165  000002                  MOV     R1,P.CYLN(R5)   ;SET CYL NO.
 4242  016122  004737  017250          12$:    JSR     PC,REDOFS       ;DO READS WITH OFSTS ON THIS CYL
 4243  016126  004737  017614                  JSR     PC,ADSCOR       ;ADD SCRATCH SCORES TO SUM
 4244  016132  005265  000002                  INC     P.CYLN(R5)      ;INCR CYL NO.
 4245  016136  005300                          DEC     R0              ;SEE IF DONE WITH THIS CYL BLK
 4246  016140  001370                          BNE     12$             ;BR IF NOT YET
 4247  016142  004737  023000                  JSR     PC,ROTBLD       ;ROTATE TABLED FOR NEXT CURRENT ZONE
 4248  016146  063701  024370                  ADD     HOLD3,R1        ;INCR CYL BY 100/200
 4249  016152  112737  000001  003147          MOVB    #1,DIF100       ;SET "CYL DIF = 100" FLAG
 4250  016160  005303                          DEC     R3              ;SEE IF DONE WITH ALL CYL BLKS
 4251  016162  001353                          BNE     10$             ;BR IF NOT YET
 4252  016164  004737  017774                  JSR     PC,AVSCOR       ;COMPUTE AVERAGES OF READ SCORES
 4253  016170  105037  003143                  CLRB    TSTTYP          ;CLEAR OFST TEST INDICATOR
 4254                                  ;ADJUST COMPAT READ SCORES (- OFSTS) BY SELF-TEST SCORES
 4255  016174  012701  007112                  MOV     #NCSCR0,R1      ;SET COMPAT SCORE POINTER
 4256  016200  012703  007412                  MOV     #NSSCR0,R3      ;SET SELF-TEST SCORE POINTER
 4257  016204  012700  005512          14$:    MOV     #DRVLST,R0      ;SET DRIVE LIST  POINTER
 4258  016210  012737  000014  001304          MOV     #12.,$TMP11     ;COMPUTE DIFFERENCE OF 12 (DEC) AND
 4259                                                                  ;    SELF-TEST SCORE FOR THIS TRACK
 4260  016216  162337  001304                  SUB     (R3)+,$TMP11
 4261  016222  010104                          MOV     R1,R4           ;GET READ SCORE POINTER IN R4
 4262  016224  005720                  15$:    TST     (R0)+           ;SEE IF A DRIVE LISTED HERE
 4263  016226  001417                          BEQ     18$             ;BR IF NOT
 4264  016230  013737  001304  001306          MOV     $TMP11,$TMP12   ;PUT DIFFERENCE INTO $TMP12
 4265  016236  061437  001306                  ADD     (R4),$TMP12     ;ADD DIFF TO READ SCORE
 4266  016242  023727  001306  000014          CMP     $TMP12,#12.     ;SEE IF 12 EXCEEDED
 4267  016250  101403                          BLOS    16$             ;BR IF NOT
 4268  016252  012737  000014  001306          MOV     #12.,$TMP12     ;SET SCORE = 12.
 4269  016260  013724  001306          16$:    MOV     $TMP12,(R4)+    ;PUT BACK ADJUSTED SCORE
 4270  016264  000757                          BR      15$             ;BR TO CONTINUE
 4271  016266  062701  000040          18$:    ADD     #32.,R1         ;POINT TO SCORES FOR NEXT TRACK
 4272  016272  020127  007212                  CMP     R1,#NCSCR0+64.  ;SEE IF SCORE TABLE EXCEEDED
 4273  016276  101742                          BLOS    14$             ;BR IF NOT YET
 4274                                  ;ADJUST COMPAT READ SCORES (+ OFSTS) BY SELF-TEST SCORES
 4275  016300  012701  007252                  MOV     #PCSCR0,R1      ;SET COMPAT SCORE POINTER
 4276  016304  012703  007420                  MOV     #PSSCR0,R3      ;SET SELF-TEST SCORE POINTER
 4277  016310  012700  005512          24$:    MOV     #DRVLST,R0      ;SET DRIVE LIST  POINTER
 4278  016314  012737  000014  001304          MOV     #12.,$TMP11     ;COMPUTE DIFFERENCE OF 12 (DEC) AND
 4279                                                                  ;    SELF-TEST SCORE FOR THIS TRACK
```

```
 4280  016322  162337  001304                    SUB     (R3)+,$TMP11
 4281  016326  010104                            MOV     R1,R4           ;GET READ SCORE POINTER IN R4
 4282  016330  005720                    25$:    TST     (R0)+           ;SEE IF A DRIVE LISTED HERE
 4283  016332  001417                            BEQ     28$             ;BR IF NOT
 4284  016334  013737  001304  001306            MOV     $TMP11,$TMP12   ;PUT DIFFERENCE INTO $TMP12
 4285  016342  061437  001306                    ADD     (R4),$TMP12     ;ADD DIFF TO READ SCORE
 4286  016346  023727  001306  000014            CMP     $TMP12,#12.     ;SEE IF 12 EXCEEDED
 4287  016354  101403                            BLOS    26$             ;BR IF NOT
 4288  016356  012737  000014  001306            MOV     #12.,$TMP12     ;SET SCORE = 12.
 4289  016364  013724  001306            26$:    MOV     $TMP12,(R4)+    ;PUT BACK ADJUSTED SCORE
 4290  016370  000757                            BR      25$             ;BR TO CONTINUE
 4291  016372  062701  000040            28$:    ADD     #32.,R1         ;POINT TO SCORES FOR NEXT TRACK
 4292  016376  020127  007352                    CMP     R1,#PCSCR0+64. ;SEE IF SCORE TABLE EXCEEDED
 4293  016402  101742                            BLOS    24$             ;BR IF NOT YET
 4294
 4295
 4296
 4297                                    ;;*****************************************************************
 4298                                    ;*TEST 11       TYPE TEST SCORES, DISMOUNT PACK, IN PASS 2
 4299                                    ;*
 4300                                    ;*      THE OVERWRITE AND COMPATIBILITY DATA READ TEST SCORES FOR THIS
 4301                                    ;*      DRIVE ARE CONVERTED AND TYPED. THEN, THE OPERATOR IS
 4302                                    ;*      DIRECTED TO UNLOAD THE DRIVE CURRENTLY UNDER TEST, AND TO REMOVE
 4303                                    ;*      THE TEST PACK. THEN, THE PROGRAM DOES 1 OF 4 THINGS :
 4304                                    ;*      (1) IF THERE IS ANOTHER DRIVE TO TEST ON THIS SUBSYS, THE
 4305                                    ;*      PROGRAM JUMPS TO TEST 5 FOR THE NEXT DRIVE.
 4306                                    ;*      (2) IF THERE IS ONLY ONE SUBSYS (FROM 200 START), AND IF
 4307                                    ;*      PASS 2 HAS BEEN COMPLETED ON ALL DRIVES, THE ENTIRE TESTING AND
 4308                                    ;*      REPORTING HAVE BEEN COMPLETED, AND THE PROGRAM TYPES A
 4309                                    ;*      COMPLETION MESSAGE, AND THEN IT RESTARTS AT ADRS 200.
 4310                                    ;*      (3) IF THERE IS ANOTHER SUBSYS TO PERFORM PASS 2 ON, THE
 4311                                    ;*      PROGRAM TELLS THE OPERATOR TO RESTART AT ADDRESS 220, AND THEN
 4312                                    ;*      IT HALTS.
 4313                                    ;*      (4) IF THERE ARE NO MORE DRIVES TO TEST ON ANY OTHER SUBSYS, THE
 4314                                    ;*      PROGRAM TYPES A COMPLETION MESSAGE AND THEN IT HALTS.
 4315                                    ;*
 4316                                    ;*****************************************************************
 4317  016404  000004                    TST11:  SCOPE
 4318  016406  012737  000011  001334            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 4319  016414  004737  022020                    JSR     PC,SETTUP       ;SET UP FOR LOOP ON ERROR
 4320  016420  004737  021140                    JSR     PC,INITSS       ;INITIALIZE DRIVER PARAMS AND SUB-SYS
 4321  016424  004737  020744                    JSR     PC,PREPKB       ;PREPARE FOR POSSIBLE KBD INPUT
 4322  016430  112737  000001  003127            MOVB    #1,TSTING       ;ALLOW TTY ESCAPE
 4323                                    ;CONVERT AND TYPE TEST SCORES
 4324  016436  013700  005554                    MOV     DRVPTR,R0       ;GET DRIVE LIST POINTER
 4325  016442  112037  010713                    MOVB    (R0)+,RESLTS+20. ;GET NO. OF THIS DRIVE
 4326  016446  111037  010712                    MOVB    (R0),RESLTS+19.
 4327  016452  104401  010667                    TYPE    ,RESLTS         ;TYPE TEST RESULT HEADINGS
 4328  016456  012737  006612  001306            MOV     #NOSCR0,$TMP12  ;GET POINTER TO OVRWRT SCORES
 4329  016464  012700  000060                    MOV     #'0,R0          ;INIT TRACK NO. TO 0
 4330  016470  012701  005512            5$:     MOV     #DRVLST,R1      ;GET DRIVE LIST POINTER
 4331  016474  110037  011175                    MOVB    R0,TRKBUF+1     ;SET TRACK NO. FOR PRINTOUT
 4332  016500  013704  001306                    MOV     $TMP12,R4       ;GET SCORE POINTER
 4333  016504  004737  025542            6$:     JSR     PC,CTLOUT       ;CHECK FOR TTY ESCAPE RQST
 4334  016510  104401  011174                    TYPE    ,TRKBUF         ;TYPE TRACK (HEAD) NUMBER
 4335  016514  012103                            MOV     (R1)+,R3        ;GET A DRIVE NAME
```

```
 4336  016516  02037?  167032                  CMP     R3,@DRVPTR      ;SEE IF IT IS CURRENT DRIVE
 4337  016522  001003                          BNE     8$              ;BR IF NOT
 4338  016524  104401  011205                  TYPE    ,SELF           ;TYPE "SELF"
 4339  016530  000407                          BR      10$             ;CONTINUE
 4340  016532  110337  011202          8$:     MOVB    R3,DRVBUF+2     ;GET DRIVE NO.
 4341  016536  000303                          SWAB    R3
 4342  016540  110337  011201                  MOVB    R3,DRVBUF+1     ;GET SUBSYS NAME
 4343  016544  104401  011200                  TYPE    ,DRVBUF         ;TYPE NO. OF DRIVE READ
 4344  016550  005037  005212          10$:    CLR     BUFF0           ;CLEAR TTY CHAR BUF
 4345  016554  005037  005214                  CLR     BUFF0+2
 4346  016560  010437  001310                  MOV     R4,$TMP13       ;GET SCORE POINTER
 4347  016564  012437  005212                  MOV     (R4)+,BUFF0     ;PUT AN OVRWRT SCORE INTO BUF
 4348  016570  012737  000004  001304          MOV     #4,$TMP11       ;SET COUNTER = 4
 4349  016576  023727  005212  000012  11$:    CMP     BUFF0,#10.      ;SEE IF SCORE < 10.
 4350  016604  103002                          BHIS    12$             ;BR IF NOT
 4351  016606  104401  012120                  TYPE    ,SPACE1         ;TYPE 1 SPACE
 4352  016612  023727  005212  000007  12$:    CMP     BUFF0,#7        ;SEE IF SCORE > 7
 4353  016620  101003                          BHI     14$             ;BR IF YES
 4354  016622  104401  012140                  TYPE    ,PROMPT         ;TYPE "* "
 4355  016626  000402                          BR      16$             ;CONTINUE
 4356  016630  104401  012117          14$:    TYPE    ,SPACE2         ;TYPE 2 SPACES
 4357  016634  012746  005212          16$:    MOV     #BUFF0,-(SP)    ;GET BUF ADRS ON STACK
 4358  016640  004737  043162                  JSR     PC,@#$DB2D      ;CONVERT SCORE TO DEC.
 4359  016644  004737  043356                  JSR     PC,@#$SUPRS     ;TYPE IT
 4360  016650  104401  011213                  TYPE    ,TAB            ;TYPE A TAB
 4361  016654  062737  000140  001310          ADD     #96.,$TMP13
 4362  016662  017737  162422  005212          MOV     @$TMP13,BUFF0   ;PUT NEXT SCORE INTO BUFFER
 4363  016670  005337  001304                  DEC     $TMP11          ;DECREMENT SCORE COUNTER
 4364  016674  001340                          BNE     11$             ;BR IF READ SCORE SHOULD BE TYPED
 4365  016676  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR>, <LF>
 4366  016702  005711                          TST     (R1)            ;SEE IF ALL SCORES TYPED FOR THIS TRACK
 4367  016704  001277                          BNE     6$              ;BR IF NOT
 4368  016706  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR>, <LF>
 4369  016712  005200                          INC     R0              ;INCR TRACK NO.
 4370  016714  062737  000040  001306          ADD     #32.,$TMP12     ;INCREMENT SCORE POINTER FOR NEXT TRACK
 4371  016722  020027  000062                  CMP     R0,#'2          ;SEE IF ALL TRACKS TYPED YET
 4372  016726  101660                          BLOS    5$              ;BR IF NOT YET
 4373                          ;COMPUTE, CONVERT, AND TYPE TEST AVERAGES (OVER ALL SURFACES)
 4374  016730  004737  020232                  JSR     PC,TRKAVG       ;GET AVGS OVER ALL TRKS
 4375  016734  013700  005554                  MOV     DRVPTR,R0       ;PUT DRIVE NAME INTO BUF
 4376  016740  112037  011262                  MOVB    (R0)+,BADDRV+7
 4377  016744  111037  011261                  MOVB    (R0),BADDRV+6
 4378  016750  005037  001304                  CLR     $TMP11          ;CLEAR SCORE INDICATOR
 4379  016754  013737  007426  005212          MOV     OVRAVG,BUFF0    ;GET OVRWRT AVG SCORE
 4380  016762  012737  011102  016776          MOV     #OAVERG,22$
 4381  016770  104401  011253          20$:    TYPE    ,BADDRV         ;TYPE "DRIVE XX"
 4382  016774  104401                          TYPE                    ;TYPE WHICH KIND OF AVERAGE
 4383  016776  000000          22$:    .WORD   0
 4384  017000  012746  005212                  MOV     #BUFF0,-(SP)    ;PUT POINTER TO AVG ON STACK
 4385  017004  004737  043162                  JSR     PC,@#$DB2D      ;CONVERT TO DECIMAL
 4386  017010  004737  043356                  JSR     PC,@#$SUPRS     ;TYPE IT
 4387  017014  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR>, <LF>
 4388  017020  013737  007430  005212          MOV     COMAVG,BUFF0    ;GET READ AVG SCORE
 4389  017026  012737  011121  016776          MOV     #RAVERG,22$
 4390  017034  005137  001304                  COM     $TMP11          ;COMPLEMENT SCORE INDIC
 4391  017040  001353                          BNE     20$             ;BR TO TYPE READ SCORE
```

```
 4392  017042  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR>,<LF>
 4393                                  ;DISMOUNT PACK ON THIS DRIVE
 4394  017046  105037  003127                  CLRB    TSTING          ;DON'T ALLOW TTY ESCAPE
 4395  017052  004737  022450                  JSR     PC,DMCART       ;DIRECT OPERATOR TO DISMOUNT PACK
 4396  017056  062737  000002  005554          ADD     #2,DRVPTR       ;INCREMENT DRIVE LIST POINTER
 4397  017064  013700  005554                  MOV     DRVPTR,R0
 4398  017070  005710                          TST     (R0)            ;SEE IF ALL DONE WITH PASS 2 YET
 4399  017072  001421                          BEQ     28$             ;BR IF DONE WITH PASS 2
 4400  017074  126037  000001  005510          CMPB    1(R0),SUBSYS    ;SEE IF DONE WITH THIS SUBSYS YET
 4401  017102  001005                          BNE     24$             ;BR IF DONE WITH THIS SUBSYS
 4402  017104  112737  000004  001102          MOVB    #4,$TSTNM       ;SET TEST NO. = 4
 4403  017112  000137  015116                  JMP     TST5            ;DO PASS 2 ON NEXT DRIVE OF SUBSYS
 4404  017116  116037  000001  010544  24$:     MOVB    1(R0),STR220+39. ;PUT SUBSYS NAME INTO MSG
 4405  017124  104401  010475                  TYPE    ,STR220         ;TYPE RESTRT MSG FOR PASS 2
 4406  017130  000000                  26$:     HALT                    ;HALT THE PROCESSOR SO THAT OPERATOR
 4407  017132  000137  000220                  JMP     220             ;  CAN RESTART AT PROPER ADRS
 4408  017136  104401  011140          28$:     TYPE    ,ENDTST         ;TYPE "**  END OF TESTING  **"
 4409  017142  012737  005512  005554          MOV     #DRVLST,DRVPTR  ;RE-INIT DRIVE LIST POINTER
 4410  017150  105737  003126                  TSTB    MDFLAG          ;SEE IF 200 START
 4411  017154  001002                          BNE     30$             ;BR IF NOT
 4412  017156  000137  000200                  JMP     200             ;RESTART AT ADRS 200
 4413  017162  000000                  30$:     HALT                    ;HALT THE PROCESSOR
 4414  017164  000137  000204                  JMP     204             ;RESTART AT ADRS 204
 4415
 4416
 4417
 4418                                  .SBTTL  END OF PASS ROUTINE
 4419
 4420                                  ;;**********************************************************************
 4421                                  ;*INCREMENT THE PASS NUMBER ($PASS)
 4422                                  ;*IF THERES A MONITOR GO TO IT
 4423                                  ;*IF THERE ISN'T JUMP TO 200
 4424
 4425  017170                          $EOP:
 4426  017170  000004                          SCOPE
 4427  017172  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 4428  017176  005237  001336                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
 4429  017202  042737  100000  001336          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 4430  017210  005327                          DEC     (PC)+           ;;LOOP?
 4431  017212  000001                  $EOPCT: .WORD   1
 4432  017214  003013                          BGT     $DOAGN          ;;YES
 4433  017216  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 4434  017220  000001                  $ENDCT: .WORD   1
 4435  017222  017212                          $EOPCT
 4436  017224  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 4437  017230  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 4438  017232  000005                          RESET                   ;;CLEAR THE WORLD
 4439  017234  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 4440  017236  000240                          NOP                     ;;SAVE ROOM
 4441  017240  000240                          NOP                     ;;FOR
 4442  017242  000240                          NOP                     ;;ACT11
 4443  017244                          $DOAGN:
 4444  017244  000137  000200                  JMP     @#200           ;;RETURN
 4445
 4446
 4447
```

```
                                        ;;*****************************************************************
                                        ;*REDOFS - READ DATA WITH RANGE OF OFFSETS
                                        ;*THIS SUBROUTINE PERFORMS READS WITH OFFSETS LISTED IN OFSTBL.
                                        ;*THE OFFSETS ARE APPLIED MINIMUM TO MAXIMUM, IN BOTH DIRECTIONS,
                                        ;*UNTIL THERE IS NO FAILURE. EACH TIME A READ ERROR OCCURS,
                                        ;*THE APPROPRIATE SCORE FOR THE DRIVE INVOLVED IS INCREMENTED NO FURTHER
                                        ;*(IN THE SCRATCH SCORE TABLE) . EACH READ IS DONE TO A PACK
                                        ;*AREA, WHOSE ADDRESS IS IN THE PARAMETER BLOCK ON ENTRY.
                                        ;;*****************************************************************
017250  104407                  REDOFS: SAVREG                          ;SAVE R0-R5
                                ;INITIALIZE SCRATCH SCORES
017252  122737  000003  003143          CMPB    #3,TSTTYP               ;SEE IF DRIVE SELF TEST
017260  001406                          BEQ     5$                      ;BR IF YES
017262  012700  007432                  MOV     #NSCOR0,R0              ;GET ADRS OF SCRATCH SCORE TABLE
017266  105020                  4$:     CLRB    (R0)+                   ;INIT A SCORE TO 0
017270  020027  007572                  CMP     R0,#NSCOR0+96.          ;SEE IF DONE YET
017274  103774                          BLO     4$                      ;BR IF NOT YET
                                ;PERFORM READS WITH OFFSET
017276  112765  000117  000001  5$:     MOVB    #SEEK,P.CMND(R5) ;SET SEEK COMMAND
017304  105737  003147                  TSTB    DIF100                  ;SEE IF CYL DIF = 100/200 (OCT)
017310  001412                          BEQ     9$                      ;BR IF NOT
017312  163765  024374  000002          SUB     HOLD5,P.CYLN(R5)        ;SEEK HALF THE CYLS
017320  004737  030124                  JSR     PC,DRVCAL
017324  063765  024374  000002          ADD     HOLD5,P.CYLN(R5)        ;SEEK THE REST OF THE WAY
017332  105037  003147                  CLRB    DIF100                  ;CLEAR "CYL DIF = 100/200 FLAG"
017336  004737  030124          9$:     JSR     PC,DRVCAL               ;PERFORM SEEK
017342  105037  003156                  CLRB    OFSDIR                  ;CLEAR OFFSET DIRECTION FLAG
017346  012703  000014                  MOV     #12.,R3                 ;SET OFFSET COUNTER FOR NEG OFSTS
017352  012700  000204                  MOV     #204,R0                 ;INIT NEG OFST VALUE TO 100 UIN
017356  105037  003157          12$:    CLRB    OFSCNT                  ;INIT OFFSET NUMBER
017362  122737  000003  003143  6$:     CMPB    #3,TSTTYP               ;SEE IF SELF-TEST
017370  001020                          BNE     15$                     ;BR IF NOT
017372  116501  000005                  MOVB    P.TRCK(R5),R1           ;GET TRACK NUMBER
017376  006301                          ASL     R1                      ;MULT BY 16.
017400  006301                          ASL     R1
017402  006301                          ASL     R1
017404  006301                          ASL     R1
017406  062701  007432                  ADD     #NSCOR0,R1              ;GET POINTER TO SCORES
017412  105737  003156                  TSTB    OFSDIR                  ;SEE WHICH OFST DIRECTION
017416  001402                          BEQ     17$                     ;BR IF NEGATIVE
017420  062701  000060                  ADD     #48.,R1                 ;POINT TO POSITIVE SCORES
017424  012704  000001          17$:    MOV     #1,R4                   ;SET COUNTER TO 1
017430  000411                          BR      14$                     ;CONTINUE
017432  012701  007432          15$:    MOV     #NSCOR0,R1              ;GET POINTER TO SCORES
017436  012704  000060                  MOV     #48.,R4                 ;SET COUNTER TO 48.
017442  105737  003156                  TSTB    OFSDIR                  ;SEE WHICH OFST DIRECTION
017446  001402                          BEQ     14$                     ;BR IF NEGATIVE
017450  012701  007512                  MOV     #PSCOR0,R1              ;GET POINTER FOR POS SCORES
017454  122137  003157          14$:    CMPB    (R1)+,OFSCNT            ;SEE IF THIS IS A PERFECT SCORE YET
017460  001002                          BNE     18$                     ;BR IF NOT
017462  105261  177777                  INCB    -1(R1)                  ;INCREMENT THIS SCORE
017466  005304                  18$:    DEC     R4                      ;DECREMENT COUNTER
017470  001371                          BNE     14$                     ;BR IF NOT DONE YET
017472  110065  000006          16$:    MOVB    R0,P.OFST(R5) ;GET AN OFFSET VALUE
017476  112765  000115  000001          MOVB    #OFFSET,P.CMND(R5) ;SET OFFSET CMND
017504  004737  030124                  JSR     PC,DRVCAL               ;PERFORM OFFSET
```

```
 4504  017510  105237  003157                    INCB    OFSCNT          ;INCR OFFSET NUMBER
 4505  017514  112765  000121  000001            MOVB    #RDDATA,P.CMND(R5) ;SET READ COMMAND
 4506  017522  004737  027456                    JSR     PC,SVPRMS       ;SAVE PARAMETERS OF TRANSFER
 4507  017526  004737  027546                    JSR     PC,TRNSFR       ;DO XFER, HANDLE DCK ERRORS
 4508  017532  112765  000177  000001            MOVB    #SUBCLR,P.CMND(R5) ;SET SUBSYS CLEAR COMMAND
 4509  017540  004737  030124                    JSR     PC,DRVCAL       ;DO A SUBSYSTEM CLEAR
 4510  017544  105065  000006                    CLRB    P.OFST(R5)      ;CLEAR OFFSET VALUE
 4511  017550  112765  000115  000001            MOVB    #OFFSET,P.CMND(R5) ;SET OFFSET COMMAND
 4512  017556  004737  030124                    JSR     PC,DRVCAL       ;PERFORM OFFSET TO 0
 4513  017562  062700  000004                    ADD     #4,R0           ;INCR OFST VALUE BY 100 UIN
 4514  017566  005303                            DEC     R3              ;DECR COUNTER
 4515  017570  001274                            BNE     6$              ;BR IF NOT DONE IN THIS DIRECTION YET
 4516  017572  012703  000014                    MOV     #12.,R3         ;SET COUNTER FOR POS SFSTS
 4517  017576  012700  000004                    MOV     #4,R0           ;INIT POS OFST TO 100 UIN
 4518  017602  105137  003156                    COMB    OFSDIR          ;COMPLEMENT OFFSET DIRECTION FLAG
 4519  017606  001263                            BNE     12$             ;BR IF POS OFSTS NOT DONE YET
 4520  017610  104410                            RESREG                  ;RESTORE R0-R5
 4521  017612  000207                            RTS     PC              ;RETURN
 4522
 4523
 4524
 4525                                    ;;*******************************************************************
 4526                                    ;*ADSCOR - ADD THE + AND - SCRATCH SCORES TO THE + AND - SCORE
 4527                                    ;*SUMS, FOR EITHER THE OVRWRT, SELF,  OR COMPAT READ TEST, AS DETERMINED
 4528                                    ;*BY "TSTTYP".
 4529                                    ;;*******************************************************************
 4530  017614  104407                    ADSCOR: SAVREG                  ;SAVE R0-R5
 4531  017616  122737  000003  003143            CMPB    #3,TSTTYP       ;SEE IF SELF-TEST
 4532  017624  001031                            BNE     2$              ;BR IF NOT
 4533  017626  113700  007432                    MOVB    NSCOR0,R0       ;ADD SELF-TEST SCORES
 4534  017632  060037  007412                    ADD     R0,NSSCR0
 4535  017636  113700  007452                    MOVB    NSCOR1,R0
 4536  017642  060037  007414                    ADD     R0,NSSCR1
 4537  017646  113700  007472                    MOVB    NSCOR2,R0
 4538  017652  060037  007416                    ADD     R0,NSSCR2
 4539  017656  113700  007512                    MOVB    PSCOR0,R0
 4540  017662  060037  007420                    ADD     R0,PSSCR0
 4541  017666  113700  007532                    MOVB    PSCOR1,R0
 4542  017672  060037  007422                    ADD     R0,PSSCR1
 4543  017676  113700  007552                    MOVB    PSCOR2,R0
 4544  017702  060037  007424                    ADD     R0,PSSCR2
 4545  017706  000430                            BR      8$              ;EXIT
 4546  017710  012700  006612            2$:     MOV     #NOSCR0,R0      ;GET POINTER TO (-) OVRWRT SCORES
 4547  017714  012705  006752                    MOV     #POSCR0,R5      ;GET POINTER TO (+) OVRWRT SCORES
 4548  017720  122737  000001  003143            CMPB    #1,TSTTYP       ;SEE IF READING OVRWRT DATA
 4549  017726  001404                            BEQ     4$              ;BR IF YES
 4550  017730  012700  007112                    MOV     #NCSCR0,R0      ;GET POINTER TO (-) COMPAT DATA SCORES
 4551  017734  012705  007252                    MOV     #PCSCR0,R5      ;GET POINTER TO (+) COMPAT DATA SCORES
 4552  017740  012704  000060            4$:     MOV     #48.,R4         ;INIT COUNTER TO 48.
 4553  017744  012701  007432                    MOV     #NSCOR0,R1      ;GET POINTER TO NEG SCRATCH SCORES
 4554  017750  012703  007512                    MOV     #PSCOR0,R3      ;GET POINTER TO POS SCRATCH SCORES
 4555  017754  112102                    6$:     MOVB    (R1)+,R2        ;GET A SCRATCH NEG SCORE
 4556  017756  060220                            ADD     R2,(R0)+        ;ADD IT TO (-) SCORE SUM
 4557  017760  112302                            MOVB    (R3)+,R2        ;GET PNTR TO POS SCRATCH SCORES
 4558  017762  060225                            ADD     R2,(R5)+        ;ADD IT TO (+) SCORE SUM
 4559  017764  005304                            DEC     R4              ;DECREMENT COUNTER
```

```
 4560  017766  001372                          BNE     6$              ;BR IF NOT DONE YET
 4561  017770  104410                  8$:     RESREG                  ;RESTORE R0-R5
 4562  017772  000207                          RTS     PC              ;RETURN
 4563
 4564
 4565
 4566                                  ;;*********************************************************************
 4567                                  ;*AVSCOR - COMPUTE THE + AND - AVERAGE SCORES FOR THE OVRWRT, COMPATIBILITY
 4568                                  ;*DATA TEST, OR SELF-TEST, AS DETERMINED BY "TSTTYP". THE SUMS IN THE APPROPRIATE
 4569                                  ;*SCORE TABLES ARE EACH DIVIDED BY THE NUMBER OF CYLINDERS READ, TO
 4570                                  ;*COMPUTE THE AVERAGES.
 4571                                  ;;*********************************************************************
 4572  017774  104407                  AVSCOR: SAVREG                  ;SAVE R0-R5
 4573  017776  122737  000003  003143          CMPB    #3,TSTTYP       ;SEE IF SELF-TEST
 4574  020004  001014                          BNE     2$              ;BR IF NOT
 4575  020006  012737  007412  001304          MOV     #NSSCR0,$TMP11  ;GET ADRS OF (-) SELF-TEST SCORES
 4576  020014  012737  007420  001310          MOV     #PSSCR0,$TMP13  ;GET ADRS OF (+) SELF-TEST SCORES
 4577  020022  012705  000030                  MOV     #24.,R5         ;SET DIVISOR = 24.
 4578  020026  012737  000003  001306          MOV     #3,$TMP12       ;SET COUNTER = 3
 4579  020034  000427                          BR      12$             ;CONTINUE
 4580  020036  012737  006612  001304  2$:     MOV     #NOSCR0,$TMP11  ;GET POINTER TO (-) OVRWRT SCORES
 4581  020044  012737  006752  001310          MOV     #POSCR0,$TMP13  ;GET POINTER TO (+) OVRWRT SCORES
 4582  020052  012705  000007                  MOV     #7.,R5          ;SET DIVISOR = 7.
 4583  020056  122737  000001  003143          CMPB    #1,TSTTYP       ;SEE IF READING OVRWRT DATA
 4584  020064  001410                          BEQ     4$              ;BR IF YES
 4585  020066  012737  007112  001304          MOV     #NCSCR0,$TMP11  ;GET POINTER TO (-) COMPAT DATA SCORES
 4586  020074  012737  007252  001310          MOV     #PCSCR0,$TMP13  ;GET POINTER TO (+) COMPAT DATA SCORES
 4587  020102  012705  000140                  MOV     #96.,R5         ;SET DIVISOR = 96.
 4588  020106  012737  000060  001306  4$:     MOV     #48.,$TMP12     ;SET COUNTER = 48.
 4589  020114  105037  003156          12$:    CLRB    OFSDIR          ;INIT FLAG FOR (-) DIRECTION
 4590  020120  005000                  6$:     CLR     R0              ;CLEAR HI BITS OF DIVIDEND AND DIVISOR
 4591  020122  005001                          CLR     R1
 4592  020124  005002                          CLR     R2
 4593  020126  005004                          CLR     R4
 4594  020130  017703  161150                  MOV     @$TMP11,R3      ;SET DIVIDEND = (-) SUM
 4595  020134  105737  003156                  TSTB    OFSDIR          ;SEE WHICH OFST DIRECTION
 4596  020140  001402                          BEQ     10$
 4597  020142  017703  161142                  MOV     @$TMP13,R3      ;SET DIVIDEND = (+) SUM
 4598  020146  004737  042664          10$:    JSR     PC,M.DPID       ;DIVIDE TO GET AVERAGES
 4599  020152  010500                          MOV     R5,R0           ;GET DIVISOR
 4600  020154  006200                          ASR     R0              ;DIVIDE IT BY 2
 4601  020156  020100                          CMP     R1,R0           ;SEE IF REMNDR > HALF DIVISOR
 4602  020160  103401                          BLO     8$              ;BR IF NOT
 4603  020162  005203                          INC     R3              ;ADD 1 TO QUOTIENT
 4604  020164  105137  003156          8$:     COMB    OFSDIR          ;COMPL DIR FLAG
 4605  020170  001406                          BEQ     14$             ;BR FOR (+)
 4606  020172  010377  161106                  MOV     R3,@$TMP11      ;PUT (-) AVG INTO TABLE
 4607  020176  062737  000002  001304          ADD     #2,$TMP11
 4608  020204  000745                          BR      6$
 4609  020206  010377  161076          14$:    MOV     R3,@$TMP13      ;PUT (+) AVG INTO TABLE
 4610  020212  062737  000002  001310          ADD     #2,$TMP13
 4611  020220  005337  001306                  DEC     $TMP12          ;DECREMENT COUNTER
 4612  020224  001333                          BNE     12$             ;BR IF NOT DONE YET
 4613  020226  104410                          RESREG                  ;RESTORE R0-R5
 4614  020230  000207                          RTS     PC              ;RETURN
 4615
```

```
 4616
 4617
 4618                                   ;;*****************************************************************
 4619                                   ;*TRKAVG - COMPUTE AVERAGES OF TEST SCORES OVER ALL SURFACES
 4620                                   ;*THIS SUBROUTINE COMPUTES THE AVERAGE OF THE OVERWRITE TEST SCORES
 4621                                   ;*OVER ALL TRACKS (SURFACES) AND PLACES THE AVERAGE IN OVRAVG.
 4622                                   ;*THEN, IT COMPUTES THE AVERAGE OF THE COMPATIBILITY DATA READ SCORES
 4623                                   ;*OVER ALL TRACKS AND PLACES THE AVERAGE IN COMAVG.
 4624                                   ;;*****************************************************************
 4625  020232  104407                  TRKAVG: SAVREG                  ;SAVE R0-R5
 4626                                   ;TAKE SUMS OF OVERWRITE AND COMPAT READ TEST SCORES OVER ALL TRACKS
 4627  020234  012700  005512                  MOV     #DRVLST,R0      ;INIT DRIVE LIST POINTER
 4628  020240  005001                          CLR     R1              ;INIT DRIVE COUNTER
 4629  020242  005037  007426                  CLR     OVRAVG          ;INIT SUMS TO 0
 4630  020246  005037  007430                  CLR     COMAVG
 4631  020252  012702  006612                  MOV     #NOSCR0,R2      ;SET POINTER TO OVRWRT SCORES
 4632  020256  012703  007112                  MOV     #NCSCR0,R3      ;SET POINTER TO READ SCORES
 4633  020262  005720                  6$:     TST     (R0)+           ;COMPUTE TOTAL NUMBER OF DRIVES
 4634  020264  001402                          BEQ     8$              ;    IN R1
 4635  020266  005201                          INC     R1
 4636  020270  000774                          BR      6$
 4637  020272  010100                  8$:     MOV     R1,R0           ;PUT NO. OF DRIVES IN R0
 4638  020274  010204                          MOV     R2,R4           ;SET POINTERS TO SCORES FOR THIS TRACK
 4639  020276  010305                          MOV     R3,R5
 4640  020300  062437  007426          9$:     ADD     (R4)+,OVRAVG    ;ADD SCORES TO SUMS
 4641  020304  066437  000136  007426          ADD     94.(R4),OVRAVG
 4642  020312  062537  007430                  ADD     (R5)+,COMAVG
 4643  020316  066537  000136  007430          ADD     94.(R5),COMAVG
 4644  020324  005300                          DEC     R0              ;SEE IF ALL SCORES ADDED FOR THIS TRACK
 4645  020326  001364                          BNE     9$              ;BR IF NOT YET
 4646  020330  062702  000040                  ADD     #32.,R2         ;POINT TO SCORES FOR NEXT TRACK
 4647  020334  062703  000040                  ADD     #32.,R3
 4648  020340  020227  006712                  CMP     R2,#NOSCR0+64.  ;SEE IF ALL TRACKS DONE YET
 4649  020344  101752                          BLOS    8$              ;BR IF NOT YET
 4650                                   ;DIVIDE TO GET AVERAGES
 4651  020346  005004                          CLR     R4              ;GET DIVISOR IN R4-R5
 4652  020350  005005                          CLR     R5
 4653  020352  060105                          ADD     R1,R5
 4654  020354  060105                          ADD     R1,R5
 4655  020356  060105                          ADD     R1,R5
 4656  020360  006305                          ASL     R5
 4657  020362  012737  007426  001304          MOV     #OVRAVG,$TMP11  ;SET POINTER TO OVRWRT SCORE SUM
 4658  020370  005037  001306                  CLR     $TMP12          ;CLEAR SCORE INDICATOR
 4659  020374  005000                  10$:    CLR     R0              ;GET DIVIDEND IN R0-R1-R2-R3
 4660  020376  005001                          CLR     R1
 4661  020400  005002                          CLR     R2
 4662  020402  017703  160676                  MOV     @$TMP11,R3
 4663  020406  004737  042664                  JSR     PC,M.DPID       ;DIVIDE TO GET AVERAGE
 4664  020412  010500                          MOV     R5,R0           ;GET DIVISOR
 4665  020414  006200                          ASR     R0              ;DIVIDE IT BY 2
 4666  020416  020100                          CMP     R1,R0           ;SEE IF REMNDR > HALF DIVISOR
 4667  020420  103401                          BLO     12$             ;BR IF NOT
 4668  020422  005203                          INC     R3              ;ADD 1 TO QUOTIENT
 4669  020424  010377  160654          12$:    MOV     R3,@$TMP11      ;STORE AVERAGE SCORE
 4670  020430  012737  007430  001304          MOV     #COMAVG,$TMP11  ;SET POINTER FOR READ SCORE SUM
 4671  020436  005137  001306                  COM     $TMP12          ;COMPLEMENT SCORE INDICATOR
```

```
 4672  020442  001354                          BNE     10$             ;BR TO COMPUTE READ SCORE AVERAGE
 4673  020444  104410                          RESREG                  ;RESTORE R0-R5
 4674  020446  000207                          RTS     PC              ;RETURN
 4675
 4676
 4677
 4678                                  ;;*******************************************************************
 4679                                  ;*NEDHDL - NON-EXISTENT DRIVE HANDLER
 4680                                  ;*THIS IS THE ABNORMAL RETURN FROM THE DRIVER,
 4681                                  ;*  WHICH IS USED WHEN NED INDICATION IS EXPECTED (AND VALID).
 4682                                  ;*    (NED = NON-EXISTENT DRIVE)
 4683                                  ;;*******************************************************************
 4684  020450  032765  010000  000020  NEDHDL: BIT     #NED,P.CS2(R5)  ;SEE IF NED ON DRIVE SELECT
 4685  020456  001002                          BNE     1$              ;BR IF NED SET
 4686  020460  000137  031472                  JMP     ERRHDL          ;GO HANDLE  OTHER ERROR
 4687  020464  010046                  1$:     MOV     R0,-(SP)        ;SAVE R0,R1
 4688  020466  010146                          MOV     R1,-(SP)
 4689  020470  012700  000001                  MOV     #1,R0           ;SET BIT POINTER
 4690  020474  005001                          CLR     R1              ;CLEAR COUNTER
 4691  020476  120137  005430          2$:     CMPB    R1,DRIVE        ;SEE IF R1 = CURRENT DRIVE NUMBER
 4692  020502  001403                          BEQ     3$              ;BR IF  =
 4693  020504  005201                          INC     R1              ;INCREMENT COUNTER
 4694  020506  006300                          ASL     R0              ;SHIFT BIT POINTER
 4695  020510  000772                          BR      2$              ;TRY AGAIN
 4696  020512  040037  005476          3$:     BIC     R0,NEWON        ;CLEAR ONLINE BIT FOR  THIS DRIVE
 4697  020516  012601                          MOV     (SP)+,R1        ;RESTORE R0,R1
 4698  020520  012600                          MOV     (SP)+,R0
 4699  020522  000137  033742                  JMP     RETNML          ;TAKE  NORMAL  RETURN
 4700
 4701
 4702
 4703
 4704
 4705                                  ;;*******************************************************************
 4706                                  ;*DCKHDL - DATA CHECK ERROR HANDLER
 4707                                  ;*THIS IS THE ABNORMAL RETURN FROM THE DRIVER, WHICH IS USED
 4708                                  ;*WHEN DCK ERROR IS EXPECTED (AND VALID).
 4709                                  ;*IF A HEADER VRC ERROR OCCURS, IT IS HANDLED, ALSO.
 4710                                  ;;*******************************************************************
 4711  020526  005037  005424          DCKHDL: CLR     RECODE          ;CLEAR ERROR FLAGS
 4712  020532  032765  000400  000034          BIT     #HVRC,P.ER(R5)  ;SEE IF HEADER VRC ERROR
 4713  020540  001404                          BEQ     4$              ;BR IF NOT
 4714  020542  052737  000004  005424          BIS     #HVRCER,RECODE  ;SET HVRC ERROR FLAG
 4715  020550  000411                          BR      8$              ;EXIT
 4716  020552  032765  100000  000034  4$:     BIT     #DCK,P.ER(R5)   ;SEE IF DCK ERROR OCCURRED
 4717  020560  001002                          BNE     6$              ;BR IF YES
 4718  020562  000137  031472                  JMP     ERRHDL          ;GO HANDLE OTHER ERROR
 4719  020566  052737  000020  005424  6$:     BIS     #DCKERR,RECODE  ;SET DCK ERROR FLAG
 4720  020574  105237  003144          8$:     INCB    DCEFLG          ;INCREMENT DCK ERROR FLAG
 4721  020600  000137  033742                  JMP     RETNML          ;TAKE NORMAL RETURN FROM DRIVER
 4722
 4723
 4724
 4725                                  ;;*******************************************************************
 4726                                  .SBTTL  KBDHDL - TTY KEYBOARD INTERRUPT HANDLER
 4727                                  ;;*******************************************************************
```

```
 4728
 4729  020604  010146                  KBDHDL: MOV     R1,-(SP)        ;SAVE R1 ON STACK
 4730  020606  017701  160334                  MOV     @$TKB,R1        ;READ A CHAR FROM KBD BUFFER
 4731  020612  042701  177600                  BIC     #177600,R1      ;MASK OUT UNUSED BITS
 4732  020616  120127  000172                  CMPB    R1,#172         ;SEE IF LOWER CASE TYPED
 4733  020622  003005                          BGT     20$             ;BR IF NOT
 4734  020624  120127  000141                  CMPB    R1,#141
 4735  020630  002402                          BLT     20$             ;BR IF NOT
 4736  020632  042701  000040                  BIC     #BIT5,R1        ;MAKE IT UPPER CASE
 4737  020636  010137  005442            20$:  MOV     R1,INTCHR       ;SAVE INPUT CHARACTER
 4738  020642  122701  000003                  CMPB    #003,R1         ;SEE IF (^C) TYPED
 4739  020646  001007                          BNE     2$              ;BR IF NOT (^C)
 4740  020650  104401  012052                  TYPE    ,CNTRLC         ;ECHO (^C)
 4741  020654  042777  000100  160262  1$:     BIC     #BIT6,@$TKS     ;CLEAR KBD INTERRUPT ENABLE BIT
 4742  020662  012601                          MOV     (SP)+,R1        ;RESTORE R1
 4743  020664  000002                          RTI                     ;RETURN
 4744  020666  122701  000032            2$:   CMPB    #032,R1         ;SEE IF (^Z) TYPED
 4745  020672  001003                          BNE     3$              ;BR IF NOT (^Z)
 4746  020674  104401  012057                  TYPE    ,CNTRLZ         ;ECHO (^Z)
 4747  020700  000765                          BR      1$              ;RETURN
 4748  020702  122701  000022            3$:   CMPB    #022,R1         ;SEE IF (^R) TYPED
 4749  020706  001003                          BNE     4$              ;BR IF NOT (^R)
 4750  020710  104401  012064                  TYPE    ,CNTRLR         ;ECHO (^R)
 4751  020714  000757                          BR      1$              ;RETURN
 4752  020716  122701  000007            4$:   CMPB    #007,R1         ;SEE IF (^G) TYPED
 4753  020722  001003                          BNE     5$              ;BR IF NOT (^G)
 4754  020724  104401  012076                  TYPE    ,CNTRLG         ;ECHO (^G)
 4755  020730  000751                          BR      1$              ;RETURN
 4756  020732  104401  005442            5$:   TYPE    ,INTCHR         ;ECHO INPUT
 4757  020736  104401  001325                  TYPE    ,$CRLF          ;DO <CR> AND <LF>
 4758  020742  000744                          BR      1$              ;RETURN
 4759
 4760
 4761
 4762                                  ;;*****************************************************************
 4763                                  .SBTTL  PREPKB - PREPARE FOR KEYBOARD INPUT
 4764                                  ;*THIS SUBROUTINE CLEARS THE KEYBOARD BUFFER AND THE
 4765                                  ;*DONE BIT, AND CLEARS THE TTY INTERRUPT INPUT WORD
 4766                                  ;*INTCHR, IN PREPARATION FOR NEW TTY INPUT. IT ALSO
 4767                                  ;*ENABLES KBD INTERRUPT.
 4768                                  ;* CALL:
 4769                                  ;*      JSR     PC,PREPKB
 4770                                  ;;*****************************************************************
 4771
 4772  020744  005077  160176          PREPKB: CLR     @$TKB           ;CLEAR KBD BUFFER AND DONE BIT
 4773  020750  005037  005442                  CLR     INTCHR          ;CLEAR TTY INPUT WORD
 4774  020754  052777  000100  160162          BIS     #BIT6,@$TKS     ;ENABLE KBD INTERRUPT
 4775  020762  000207                          RTS     PC              ;RETURN
 4776
 4777
 4778
 4779                                  ;;*****************************************************************
 4780                                  .SBTTL  ECOBAD - ECHO BAD TTY INPUT
 4781                                  ;*THIS SUBPOUTINE ECHOS A CHARACTER WHICH HAD BEEN
 4782                                  ;*TYPED IN AND WAS DETERMINED TO BE INVALID FOR SOME
 4783                                  ;*REASON, FOLLOWED BY A QUESTION MARK (?). THEN, A <CR>
```

```
                                        ;*AND <LF> ARE DONE.
                                        ;* CALL -     JSR      PC,ECOBAD
                                        ;;*****************************************************************

020764  104401  005442        ECOBAD: TYPE    ,INTCHR         ;ECHO BAD CHARACTER
020770  104401  001324                TYPE    ,$QUES          ;TYPE <?> AND <CR>, <LF>
020774  000207                      RTS     PC              ;RETURN
```

```
                                        ;;*****************************************************************
                                        ;* GETPRM - INPUT A SIX-DIGIT OCTAL NO. ON TTY KBD
                                        ;*ENTER SUBROUTINE WITH OLD PARAMETER VALUE ON STACK. SUBROUTINE
                                        ;*TYPES OLD VALUE, AND INPUTS NEW VALUE, RETURNING NEW VALUE ON
                                        ;*TOP OF STACK.
                                        ;;*****************************************************************
020776  016646  000002        GETPRM: MOV     2(SP),-(SP)     ;GET OLD VALUE
021002  104403                      TYPOS                   ;TYPE IT
021004     006                      .BYTE   6               ;SIX DIGITS
021005     000                      .BYTE   0               ;SUPPRESS LEADING ZEROS
021006  104401  011224              TYPE    ,NEWMSG         ;TYPE    '' NEW = ''
021012  004737  023402              JSR     PC,RDCHRS       ;READ NEW VALUE FROM KBD
021016  021050                      7$                      ;(^C) RETURN ADDRESS
021020  021050                      7$                      ;(^Z) RETURN ADDRESS
021022  021050                      7$                      ;(^U) OR ERROR RETURN ADDRESS
021024  005700                      TST     R0              ;SEE IF ANY CHARS TYPED
021026  001001                      BNE     4$              ;BR IF YES
021030  000207                      RTS     PC              ;RETURN - OLD VALUE UNCHANGED
021032  020027  000006        4$:     CMP     R0,#6           ;SEE IF > 6 CHARS TYPED
021036  003407                      BLE     8$              ;BR IF NOT BAD
021040  104401  005212        6$:     TYPE    ,BUFF0          ;ECHO BAD INPUT
021044  104401  001324                TYPE    ,$QUES
021050  162716  000010        7$:     SUB     #10,(SP)        ;FIX ERROR RETURN PC
021054  000207                      RTS     PC              ;ERROR RETURN
021056  012746  005212        8$:     MOV     #BUFF0,-(SP)    ;PUT POINTER TO CHARS ON STACK
021062  004737  042104              JSR     PC,OCTBIN       ;CONVERT DIGITS TO BINARY
021066  021040                      6$                      ;ERROR RETURN ADDRESS
021070  012600                      MOV     (SP)+,R0        ;GET NEW BINARY VALUE
021072  005737  042236              TST     $HIOCT          ;SEE IF HI BITS ARE 0
021076  001360                      BNE     6$              ;BR IF NOT
021100  010066  000002              MOV     R0,2(SP)        ;PUT NEW VALUE ON STACK
021104  000207                      RTS     PC              ;RETURN
```

```
                                        ;;*****************************************************************
                                        .SBTTL  GTSWRG - OPEN SOFTWARE SWITCH REGISTER FOR MODIFICATION
                                        ;*THIS SUBROUTINE ALLOWS THE CONTENTS OF THE SOFTWARE SWITCH
                                        ;*REGISTER TO BE MODIFIED BY TTY INPUT. THE SUBROUTINE TYPES
                                        ;* ''SWR = XXXXXX  NEW = '', AND WAITS FOR A NEW OCTAL VALUE
                                        ;*OF UP TO SIX DIGITS TO BE TYPED.
                                        ;;*****************************************************************
021106  022737  000176  001140  GTSWRG: CMP     #SWREG,SWR      ;SEE IF SOFTWARE SWR SELECTED
```

```
 4840  021114  001010                          BNE     6$                ;BR IF NOT
 4841  021116  013746  000176                  MOV     SWREG,-(SP)       ;PUT OLD VALUE ON STACK
 4842  021122  104401  011215                  TYPE    ,SWRMSG           ;TYPE "SWR = "
 4843  021126  004737  020776                  JSR     PC,GETPRM         ;TYPE OLD, GET NEW SWREG VALUE
 4844  021132  012637  000176                  MOV     (SP)+,SWREG       ;STORE NEW VALUE
 4845  021136  000207                  6$:     RTS     PC                ;RETURN
 4846
 4847
 4848
 4849
 4850
 4851
 4852
 4853
 4854                                  ;;*****************************************************************
 4855                                  .SBTTL  INITSS - INITIALIZE SUBSYSTEM
 4856                                  ;*
 4857                                  ;*THIS SUBROUTINE INITIALIZES THE DRIVER AND ITS PARAMETERS
 4858                                  ;*AND DOES A SUBSYSTEM CLEAR.
 4859                                  ;* USES -   R2,R5
 4860                                  ;* CALL:
 4861                                  ;*              JSR     PC,INITSS
 4862                                  ;;*****************************************************************
 4863
 4864  021140  012737  031472  003056  INITSS: MOV     #ERRHDL,A.ABNL  ;SET UP ABNORMAL ERROR RETURN ADDRESS
 4865  021146  012737  030250  003054          MOV     #ERRFRE,A.NORM  ;SET UP NORMAL RETURN ADDRESS
 4866  021154  013702  003046                  MOV     RKBAS,R2        ;GET ADDRESS OF RK611 REGISTERS
 4867  021160  012705  002640                  MOV     #PARM0,R5       ;GET ADDRESS OF PARAMETER BLOCK
 4868  021164  105037  003150                  CLRB    NORTRY          ;CLEAR "NO-RETRY" FLAG
 4869  021170  004737  021246                  JSR     PC,CLRPRM       ;CLEAR DRIVER INPUT PARAMETERS
 4870  021174  112765  000177  000001          MOVB    #SUBCLR,P.CMND(R5) ;SET SUBSYSTEM CLEAR COMMAND
 4871  021202  004737  030124                  JSR     PC,DRVCAL       ;DO SUBSYSTEM CLEAR
 4872  021206  113765  005430  000000          MOVB    DRIVE,P.DRVN(R5) ;SET CURRENT DRIVE NO.
 4873  021214  113737  005430  002724          MOVB    DRIVE,PARM1     ;SET DRIVE NO. IN ALTERNATE P.B.
 4874  021222  113765  003134  000007          MOVB    FORMAT,P.CS1H(R5) ;SET CURRENT DRIVE FORMAT
 4875  021230  153765  003133  000007          BISB    TYPFMT,P.CS1H(R5)       ;SET DRV TYPE. 0=RK06, 4=RK07
 4876  021236  116537  000007  002733          MOVB    P.CS1H(R5),P.CS1H+PAR   ;COPY FORMAT & DR TYPE INTO PAR 1
 4877  021244  000207                          RTS     PC              ;SUBROUTINE EXIT
 4878
 4879
 4880
 4881                                  ;;*****************************************************************
 4882                                  .SBTTL  CLRPRM - CLEAR DRIVER INPUT PARAMETERS
 4883                                  ;*THIS SUBROUTINE ZEROS THE FIRST 14 BYTES IN THE DRIVER
 4884                                  ;*PARAMETER BLOCK (WHOSE ADDRESS IS IN R5).
 4885                                  ;* CALL -        JSR     PC,CLRPRM
 4886                                  ;;*****************************************************************
 4887
 4888  021246  010046                  CLRPRM: MOV     R0,-(SP)        ;SAVE R0
 4889  021250  010546                          MOV     R5,-(SP)        ;SAVE R5
 4890  021252  010500                          MOV     R5,R0           ;GET PARAMETER BLOCK ADDRESS
 4891  021254  062705  000016                  ADD     #P.CS1,R5       ;COMPUTE LIMIT ADDRESS
 4892  021260  005020                  1$:     CLR     (R0)+           ;CLEAR A WORD IN PARAMETER BLOCK
 4893  021262  020005                          CMP     R0,R5           ;SEE IF DONE YET
 4894  021264  001375                          BNE     1$              ;BR IF  NOT DONE  YET
 4895  021266  012605                          MOV     (SP)+,R5        ;RESTORE R5
```

```
 4896  021270  012600                          MOV     (SP)+,R0        ;RESTORE R0
 4897  021272  000207                          RTS     PC              ;RETURN
 4898
 4899
 4900
 4901                                  ;;*****************************************************************
 4902                                  .SBTTL  SCNDRV - SCAN DRIVE FOR STATUS
 4903                                  ;*THIS SUBROUTINE SCANS THE SELECTED DRIVE TO INSURE
 4904                                  ;*THAT THE DRIVE IS ON-LINE, READY, NOT WRITE-LOCKED,
 4905                                  ;*AND NOT LOADED WITH AN ALIGNMENT CARTRIDGE. IF ANY
 4906                                  ;*OF THESE CONDITIONS ARE NOT MET, AN APPROPRIATE
 4907                                  ;*MESSAGE IS TYPED, RK06 INTERRUPT IS DISABLED, AND
 4908                                  ;*A RETURN IS MADE TO THE ADDRESS LISTED IMMEDIATELY
 4909                                  ;*AFTER THE CALL TO SCNDRV. THE REQUESTED DRIVE NO.
 4910                                  ;*MUST BE PASSED TO THE SUBROUTINE IN WORD "DRIVE".
 4911                                  ;*  CALL -    JSR     PC,SCNDRV
 4912                                  ;*            <ERROR RETURN ADDRESS>
 4913                                  ;*
 4914                                  ;;*****************************************************************
 4915
 4916  021274  004737  021140          SCNDRV: JSR     PC,INITSS       ;INITIALIZE DRIVER AND CLEAR SUBSYSTEM
 4917  021300  042712  000100          3$:     BIC     #IE,(R2)        ;DISABLE RK06 INTERRUPT
 4918  021304  113762  005430  000010          MOVB    DRIVE,RKCS2(R2) ;SET DRIVE NO. IN RKCS2
 4919  021312  105737  003133                  TSTB    TYPFMT          ;SEE IF RK07
 4920  021316  001003                          BNE     5$              ;BR IF YES
 4921  021320  012712  000001                  MOV     #GO,(R2)        ;ELSE SET GO BIT FOR RK06 IN RKCS1
 4922  021324  000402                          BR      7$
 4923
 4924  021326  012712  002001          5$:     MOV     #<CDT!GO>,(R2)  ;SET GO FOR RK07
 4925  021332  005037  005444          7$:     CLR     SCRACH          ;CLEAR STALL COUNTER
 4926  021336  005237  005444          15$:    INC     SCRACH          ;INCREMENT STALL COUNTER
 4927  021342  001375                          BNE     15$             ;STALL FOR SEVERAL MILLI-SEC
 4928  021344  032762  001000  000010          BIT     #MDS,RKCS2(R2)  ;SEE IF MDS ERROR
 4929  021352  001417                          BEQ     18$             ;BR IF NOT
 4930  021354  112765  000101  000001          MOVB    #SELDRV,P.CMND(R5) ;SET CMND FOR ERROR REPORT
 4931  021362  011265  000016                  MOV     (R2),P.CS1(R5)  ;GET RKCS1 FOR REPORT
 4932  021366  004737  037632                  JSR     PC,I.CST1       ;GET OTHER RK611 REGS FOR REPORT
 4933  021372  004737  031212                  JSR     PC,REPSUP       ;SET UP ERROR REPORT
 4934  021376  104052                          ERROR   52              ;REPORT MDS ERROR
 4935  021400  052737  000200  005424          BIS     #ABORT,RECODE   ;SET ABORT FLAG
 4936  021406  000137  033324                  JMP     ALLTRM          ;GO ABORT TESTING
 4937  021412  004737  021140          18$:    JSR     PC,INITSS       ;INIT THE S.S.
 4938  021416  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;SET READ DRIVE STATUS COMMAND
 4939  021424  012737  020450  003056          MOV     #NEDHDL,A.ABNL  ;SET NED ABNORMAL RETURN ADDRESS
 4940  021432  012737  000377  005476          MOV     #377,NEWON      ;INIT. ON-LINE INDICATOR
 4941  021440  004737  030124                  JSR     PC,DRVCAL       ;READ ALL DRIVE STATUS
 4942  021444  112737  000040  011262          MOVB    #40,BADDRV+7    ;PUT A SPACE IN MSG BUF
 4943                                  ;SEE IF THIS DRIVE EXISTS AND IS ON-LINE
 4944  021452  012737  031472  003056          MOV     #ERRHDL,A.ABNL  ;RESTORE ABNL RETURN ADDRESS
 4945  021460  022737  000377  005476          CMP     #377,NEWON      ;SEE IF NED INDICATION ON THIS DRIVE
 4946  021466  001425                          BEQ     4$              ;BR IF NED NOT SET
 4947  021470  113737  005430  011261          MOVB    DRIVE,BADDRV+6  ;GET DRIVE NO. INTO BUF
 4948  021476  152737  000060  011261          BISB    #'0,BADDRV+6    ;CONVERT TO ASCII
 4949  021504  104401  011253                  TYPE    ,BADDRV         ;TYPE "DRIVE X"
 4950  021510  104401  011264                  TYPE    ,NXDRIV         ;TYPE "NON-EXISTENT"
 4951  021514  132737  000200  001351          BITB    #BIT7,$ENVM     ;SEE IF UNDER APT
```

```
 4952  021522  001401                          BEQ     2$              ;BR IF NO
 4953  021524  104123                          ERROR   123             ;NED UNDER APT SIZING
 4954                                ;SERVICE ERRORS HERE
 4955  021526  042762  000100  000000  2$:     BIC     #IE,RKCS1(R2)   ;DISABLE RK06 INTERRUPT
 4956  021534  017616  000000                  MOV     @(SP),(SP)      ;SET UP ERROR RETURN ADDRESS
 4957  021540  000207                          RTS     PC              ;ERROR RETURN
 4958                                ;SEE IF DRIVE IS READY
 4959  021542  032765  000200  000040  4$:     BIT     #S.DRY,P.A00(R5) ;TEST FOR DRIVE READY
 4960  021550  001013                          BNE     6$              ;BR IF DRIVE IS READY
 4961  021552  113737  005430  011261          MOVB    DRIVE,BADDRV+6  ;GET DRIVE NO. INTO BUF
 4962  021560  152737  000060  011261          BISB    #'0,BADDRV+6    ;CONVERT TO ASCII
 4963  021566  104401  011253                  TYPE    ,BADDRV         ;TYPE "DRIVE X"
 4964  021572  104401  011303                  TYPE    ,NTREDY         ;TYPE "NOT READY"
 4965  021576  000753                          BR      2$              ;TAKE ERROR EXIT
 4966                                ;SEE IF DRIVE IS WRITE ENABLED
 4967  021600  032765  004000  000040  6$:     BIT     #S.WRL,P.A00(R5) ;SEE IF WRITE LOCK SET
 4968  021606  001413                          BEQ     8$              ;BR IF WRITE LOCK NOT SET
 4969  021610  113737  005430  011261          MOVB    DRIVE,BADDRV+6  ;GET DRIVE NO.
 4970  021616  152737  000060  011261          BISB    #'0,BADDRV+6    ;CONVERT TO ASCII
 4971  021624  104401  011253                  TYPE    ,BADDRV         ;TYPE "DRIVE X"
 4972  021630  104401  011317                  TYPE    ,WRTLOK         ;TYPE "WRITE-LOCKED"
 4973  021634  000734                          BR      2$              ;TAKE ERROR EXIT
 4974                                ;SEE IF DRIVE NOT LOADED WITH ALIGNMENT CARTRIDGE
 4975  021636  112765  000103  000001  8$:     MOVB    #PACK,P.CMND(R5) ;SET PACK ACKNOWLEDGE COMMAND
 4976  021644  004737  030124                  JSR     PC,DRVCAL       ;SET VOLUME VALID
 4977  021650  112765  000113  000001          MOVB    #RECAL,P.CMND(R5)       ;SET RECAL CMND
 4978  021656  004737  030124                  JSR     PC,DRVCAL       ;RECAL THE DRIVE
 4979  021662  112765  000121  000001          MOVB    #RDDATA,P.CMND(R5) ;SET READ COMMAND
 4980  021670  013765  024362  000002          MOV     LSTCYL,P.CYLN(R5) ;SET CYLINDER = 632(8)/1456(8)
 4981  021676  112765  000002  000005          MOVB    #LSTTRK,P.TRCK(R5) ;SET TRACK = 2
 4982  021704  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;BUS ADDRESS
 4983  021712  012765  177774  000012          MOV     #-4,P.WC(R5)    ;READ 4 WORDS
 4984  021720  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5)      ;SET 22 SECTOR FORMAT
 4985  021726  153765  003133  000007          BISB    TYPFMT,P.CS1H(R5)
 4986  021734  004737  030124                  JSR     PC,DRVCAL       ;READ 4 WORDS OF BAD SECTOR FILE
 4987  021740  032737  100000  005424          BIT     #ANYDER,RECODE  ;SEE IF DATA ERROR
 4988  021746  001402                          BEQ     10$             ;BR IF OK
 4989  021750  104401  012006                  TYPE    ,BAD632         ;TYPE READ ERROR MESSAGE
 4990  021754  022737  177777  053336  10$:     CMP     #177777,RWBUF+6 ;SEE IF ALL 1'S IN I.D. WORD 3
 4991  021762  001013                          BNE     12$             ;BR IF NOT ALL 1'S
 4992  021764  113737  005430  011261          MOVB    DRIVE,BADDRV+6  ;GET DRIVE NO.
 4993  021772  152737  000060  011261          BISB    #'0,BADDRV+6    ;CONVERT TO ASCII
 4994  022000  104401  011253                  TYPE    ,BADDRV         ;TYPE "DRIVE X"
 4995  022004  104401  011336                  TYPE    ,ALNPAK         ;TYPE "LOADED WITH ALIGN PACK"
 4996  022010  000646                          BR      2$              ;TAKE ERROR EXIT
 4997                                ;ERROR FREE RETURN
 4998  022012  062716  000002          12$:    ADD     #2,(SP)         ;FIX UP RETURN PC
 4999  022016  000207                          RTS     PC              ;RETURN
 5000
 5001
 5002                                ;;*****************************************************************
 5003                                ;*SETTUP - SET UP FOR LOOP ON ERROR
 5004                                ;*THIS SUBROUTINE CANNOT BE CALLED BY ANY OTHER
 5005                                ;*SUBROUTINE --- ONLY MAIN-LINE CODE !!!!
 5006                                ;;*****************************************************************
 5007  022020  011637  001076          SETTUP: MOV     (SP),@#STACK-2  ;MOVE RETURN PC ON STACK
```

```
 5008  022024  012706  001076                  MOV     #STACK-2,SP     ;RE-INIT THE STACK POINTER
 5009  022030  005037  005424                  CLR     RECODE          ;CLEAR ERROR RECOVERY FLAGS
 5010  022034  105037  003135                  CLRB    ERRCNT          ;CLEAR RETRY ERROR COUNT
 5011  022040  012705  002640                  MOV     #PARM0,R5       ;SET PARAM BLK ADRS
 5012  022044  112765  000177  000001          MOVB    #SUBCLR,P.CMND(R5) ;SET SUBSYSTEM CLEAR CMND
 5013  022052  004737  030124                  JSR     PC,DRVCAL       ;CLEAR THE SUBSYSTEM
 5014  022056  012737  000000  177776          MOV     #PR0,@#PS       ;RE-ESTABLISH PRIORITY 0
 5015  022064  000207                          RTS     PC              ;RETURN
 5016
 5017
 5018
 5019
 5020
 5021                                  ;;*****************************************************************
 5022                                  .SBTTL  MTCART - MOUNT TEST CARTRIDGE, CHECK DRIVE STATUS
 5023                                  ;*THIS SUBROUTINE PERFORMS THE FOLLOWING FUNCTIONS :
 5024                                  ;*IT TYPES "STARTING PASS X", DIRECTS THE OPERATOR TO MOUNT THE
 5025                                  ;*TEST CARTRIDGE, CHECKS THE DRIVE STATUS, DETERMINES THE PACK FORMAT,
 5026                                  ;*READS THE BAD SECTOR FILES, AND TYPES THE DRIVE AND CART. SERIAL
 5027                                  ;*NUMBERS.
 5028                                  ;;*****************************************************************
 5029  022066                          MTCART:
 5030  022066  013700  005554                  MOV     DRVPTR,R0       ;GET POINTER TO FIRST DRIVE FOR THIS SUBSYS
 5031  022072  112037  010336                  MOVB    (R0)+,MNTPAK+22. ;PUT DRIVE NAME INTO MSG
 5032  022076  111037  010335                  MOVB    (R0),MNTPAK+21.
 5033  022102  162700  005512                  SUB     #DRVLST,R0      ;SUBTRACT LIST ADRS FROM POINTER
 5034  022106  006200                          ASR     R0              ;DIVIDE BY TWO TO GET DRIVE NO.
 5035  022110  110037  003154                  MOVB    R0,LOGDRV       ;STORE LOGICAL DRIVE NUMBER
 5036  022114  004737  021140                  JSR     PC,INITSS       ;INITIALIZE SUBSYSTEM
 5037  022120  042762  000100  000000          BIC     #IE,RKCS1(R2)   ;DISABLE RK06 INTERRUPT FOR MAN. INTERVENTION
 5038  022126  104401  010310          10$:    TYPE    ,MNTPAK         ;TYPE MOUNT PACK MSG
 5039  022132  104401  010552          5$:     TYPE    ,TYPRWN         ;TYPE "TYPE R<CR> WHEN "
 5040  022136  104401  010573                  TYPE    ,DRIRDY         ;TYPE "DRIVE READY :"
 5041  022142  004737  023402                  JSR     PC,RDCHRS       ;WAIT FOR R<CR> RESPONSE
 5042  022146  022172                          12$                     ;(^C) RETURN ADDRESS
 5043  022150  022214                          16$                     ;(^Z) RETURN ADDRESS
 5044  022152  022132                          5$                      ;(^U) RETURN ADDRESS
 5045  022154  005700                          TST     R0              ;SEE IF NULL INPUT
 5046  022156  001025                          BNE     8$              ;BR IF NOT NULL
 5047  022160  104401  005212          6$:     TYPE    ,BUFF0          ;ECHO BAD INPUT
 5048  022164  104401  001324                  TYPE    ,$QUES          ;TYPE <?> AND <CR>,<LF>
 5049  022170  000760                          BR      5$              ;GO ASK AGAIN
 5050  022172  012706  001100          12$:    MOV     #STACK,SP       ;RE-INIT THE STACK
 5051  022176  105737  003126                  TSTB    MDFLAG          ;SEE IF 200 START
 5052  022202  001002                          BNE     14$             ;BR IF NOT
 5053  022204  000137  012146                  JMP     DFSTRT          ;RESTART PROGRAM - 200 START
 5054  022210  000137  012752          14$:    JMP     ALLSYS          ;GO ASK FOR SUBSYS'S AND DRIVES AGAIN
 5055  022214  105737  003126          16$:    TSTB    MDFLAG          ;SEE IF 200 START
 5056  022220  001744                          BEQ     5$              ;BR IF YES, TO ASK AGAIN
 5057  022222  012706  001100                  MOV     #STACK,SP       ;RE-INIT THE STACK
 5058  022226  000137  013416                  JMP     ASKSYS          ;GO ASK FOR SUBSYS TO TEST AGAIN
 5059  022232  023727  005212  000122  8$:     CMP     BUFF0,#'R       ;SEE IF R<CR> TYPED
 5060  022240  001347                          BNE     6$              ;BR IF NOT
 5061  022242  004737  021140                  JSR     PC,INITSS       ;INITIALIZE S.S.
 5062                                  ;CHECK STATUS OF DRIVE TO BE TESTED NEXT
 5063  022246  117737  163302  005430          MOVB    @DRVPTR,DRIVE   ;SET DRIVE NUMBER
```

```
 5064  022254  142737  000060  005430          BICB    #'0,DRIVE       ;STRIP THE ASCII
 5065  022262  004737  021274                  JSR     PC,SCNDRV       ;CHECK STATUS OF THIS DRIVE -
 5066  022266  022126                          10$                     ;  IF NOT VALID, TYPE MSG AND RETURN
 5067                                                                  ;  TO WAIT FOR R<CR> AGAIN
 5068                                  ;READ THE HEADER ON SECTOR 0,TRACK 0,CYL 0, AND GET THE DRIVE FORMAT
 5069  022270  004737  021140                  JSR     PC,INITSS       ;INIT. DRIVER PARAMS AND S.S.
 5070  022274  112765  000125  000001          MOVB    #RDHEAD,P.CMND(R5) ;SET READ HEADER COMMAND
 5071  022302  004737  030124                  JSR     PC,DRVCAL       ;DO A READ HEADER
 5072  022306  013762  024364  000000          MOV     HOLD1,RKCS1(R2) ;DO A READ HDR, WITH FMT BIT = 0
 5073  022314  032762  000200  000000  24$:    BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
 5074  022322  001774                          BEQ     24$
 5075  022324  013762  024366  000000          MOV     HOLD2,RKCS1(R2) ;DO A READ HDR, WITH FMT BIT = 1
 5076  022332  032762  000200  000000  26$:    BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
 5077  022340  001774                          BEQ     26$
 5078  022342  105037  003134                  CLRB    FORMAT          ;INITIALIZE FORMAT BYTE TO 0
 5079  022346  005762  000024                  TST     RKDB(R2)        ;POP SILO ONE TIME
 5080  022352  032762  001000  000024          BIT     #BIT9,RKDB(R2)  ;TEST FORMAT BIT IN HEADER WORD 2
 5081  022360  001403                          BEQ     28$             ;BR IF 22 SECTOR FORMAT
 5082  022362  152737  000020  003134          BISB    #B.CFMT,FORMAT  ;SET 20 SECTOR FORMAT FOR THIS DRIVE
 5083  022370  004737  021140          28$:    JSR     PC,INITSS       ;INIT THE S.S.
 5084  022374  004737  026516                  JSR     PC,REDBSF       ;READ BAD SECTOR FILE FOR THIS DRIVE
 5085  022400  105737  003126                  TSTB    MDFLAG          ;SEE IF 200 START
 5086  022404  001004                          BNE     32$             ;BR IF NOT 200 START
 5087  022406  122737  000061  003155          CMPB    #'1,PASSNO      ;SEE IF FIRST PASS
 5088  022414  001012                          BNE     30$             ;BR IF NOT FIRST PASS
 5089  022416  004737  023714          32$:    JSR     PC,DRVSER       ;TYPE "DRIVE SER. NO.  XXX"
 5090  022422  004737  024034                  JSR     PC,CRTSER       ;TYPE "CART. SER. NO.  XXXXXXXXXXX"
 5091  022426  032777  000200  156504          BIT     #BIT7,@SWR      ;SEE IF BAD SECTORS SHOULD BE TYPED
 5092  022434  001402                          BEQ     30$             ;BR IF NOT
 5093  022436  004737  027006                  JSR     PC,TYPBSF       ;TYPE BAD SECTOR FILES
 5094  022442  005037  005442          30$:    CLR     INTCHR          ;INIT. TTY INPUT CHAR BUFFER
 5095  022446  000207                          RTS     PC              ;RETURN
 5096
 5097
 5098
 5099                                  ;;*****************************************************************
 5100                                  .SBTTL  DMCART - DISMOUNT TEST CARTRIDGE
 5101                                  ;*THIS SUBROUTINE DIRECTS THE OPERATOR IN THE DISMOUNTING
 5102                                  ;*OF THE TEST CARTRIDGE ON THE DRIVE CURRENTLY UNDER TEST.
 5103                                  ;;*****************************************************************
 5104  022450                          DMCART:
 5105  022450  013700  005554                  MOV     DRVPTR,R0       ;GET DRIVE LIST POINTER
 5106  022454  112037  010373                  MOVB    (R0)+,DISPAK+15. ;PUT DRIVE NAME INTO MSG
 5107  022460  111037  010372                  MOVB    (R0),DISPAK+14.
 5108  022464  004737  021140                  JSR     PC,INITSS       ;INITIALIZE SUBSYS
 5109  022470  042762  000100  000000          BIC     #IE,RKCS1(R2)   ;DISABLE RK06 INTRPT FOR MANUAL INTERVENTION
 5110  022476  104401  010354                  TYPE    ,DISPAK         ;TYPE "UNLOAD DRIVE XX AND REMOVE PACK"
 5111  022502  104401  010552          5$:     TYPE    ,TYPRWN         ;TYPE "TYPE R<CR> WHEN "
 5112  022506  104401  010612                  TYPE    ,DRIDON         ;TYPE "DONE :"
 5113  022512  004737  023402                  JSR     PC,RDCHRS       ;WAIT FOR R<CR> RESPONSE
 5114  022516  022542                          12$                     ;(^C) RETURN ADDRESS
 5115  022520  022564                          16$                     ;(^Z) RETURN ADDRESS
 5116  022522  022502                          5$                      ;(^U) RETURN ADDRESS
 5117  022524  005700                          TST     R0              ;SEE IF NULL INPUT
 5118  022526  001025                          BNE     8$              ;BR IF NOT NULL
 5119  022530  104401  005212          6$:     TYPE    ,BUFFO          ;ECHO BAD INPUT
```

```
 5120  022534  104401  001324                  TYPE    ,$QUES          ;TYPE <?> AND <CR>,<LF>
 5121  022540  000760                          BR      5$              ;GO ASK AGAIN
 5122  022542  012706  001100          12$:    MOV     #STACK,SP       ;RE-INIT THE STACK
 5123  022546  105737  003126                  TSTB    MDFLAG          ;SEE IF 200 START
 5124  022552  001002                          BNE     14$             ;BR IF NOT
 5125  022554  000137  012146                  JMP     DFSTRT          ;RESTART PROGRAM - 200 START
 5126  022560  000137  012752          14$:    JMP     ALLSYS          ;GO ASK FOR SUBSYS'S AND DRIVES AGAIN
 5127  022564  105737  003126          16$:    TSTB    MDFLAG          ;SEE IF 200 START
 5128  022570  001744                          BEQ     5$              ;BR IF YES, TO ASK AGAIN
 5129  022572  012706  001100                  MOV     #STACK,SP       ;RE-INIT THE STACK
 5130  022576  000137  013416                  JMP     ASKSYS          ;GO ASK FOR SUBSYS TO TEST AGAIN
 5131  022602  023727  005212  000122  8$:     CMP     BUFFO,#'R       ;SEE IF R<CR> TYPED
 5132  022610  001347                          BNE     6$              ;BR IF NOT
 5133  022612  004737  021140                  JSR     PC,INITSS       ;INITIALIZE S.S.
 5134  022616  000207                          RTS     PC              ;RETURN
 5135
 5136
 5137
 5138                                  ;;*****************************************************************
 5139                                  .SBTTL  INTBLD - INITIALIZE TABLED
 5140                                  ;*THIS SUBROUTINE LOADS TABLED WITH THE INITIAL PATTERN OF DRIVE NUMBERS
 5141                                  ;*WHICH IS THE CYLINDER BLOCK LAYOUT FOR THE FIRST CURRENT ZONE. IT USES
 5142                                  ;*THE STARTING DRIVE NUMBER LIST, STDLST, FOR THE FIRST COLUMN, AND
 5143                                  ;*INCREMENTS THESE DRIVE NUMBERS ACROSS THE TABLE, FORMING ROWS. FOR
 5144                                  ;*ANY DRIVE WHICH IS NOT PRESENT, THE ENTRY IS SET = 377.
 5145                                  ;;*****************************************************************
 5146  022620  104407                  INTBLD: SAVREG                  ;SAVE R0-R5
 5147                                  ;LOAD THE FIRST COLUMN OF TABLED FROM STDLST
 5148  022622  012700  006552                  MOV     #STDLST,R0      ;STARTING ADRS OF STDLST
 5149  022626  012701  006112                  MOV     #TABLED,R1      ;STARTING ADRS OF TABLED
 5150  022632  012703  000020                  MOV     #16.,R3         ;SET COUNTER =  16
 5151  022636  112011                  4$:     MOVB    (R0)+,(R1)      ;LOAD A BYTE INTO TABLED
 5152  022640  062701  000020                  ADD     #16.,R1         ;INCREMENT TABLED POINTER
 5153  022644  005303                          DEC     R3              ;DECR COUNTER
 5154  022646  001373                          BNE     4$              ;BR IF NOT DONE YET
 5155                                  ;GET INDEX OF FIRST ZERO ENTRY IN DRVLST INTO R1
 5156  022650  012700  005512                  MOV     #DRVLST,R0      ;STARTING ADRS OF DRVLST
 5157  022654  005001                          CLR     R1              ;INIT INDEX TO 0
 5158  022656  005720                  6$:     TST     (R0)+           ;SEE IF ENTRY =  0
 5159  022660  001404                          BEQ     8$              ;BR IF YES
 5160  022662  005201                          INC     R1              ;INCREMENT POINTER
 5161  022664  020127  000020                  CMP     R1,#16.         ;SEE IF DONE LOOKING YET
 5162  022670  002772                          BLT     6$              ;BR IF NOT DONE YET
 5163                                  ;INCREMENT THE DRIVE NOS. TO FORM THE ROWS
 5164  022672  012700  006112          8$:     MOV     #TABLED,R0      ;INIT TABLED POINTER
 5165  022676  012704  000017          16$:    MOV     #15.,R4         ;INIT ROW ELEMENT COUNTER
 5166  022702  112003                          MOVB    (R0)+,R3        ;INIT ROW ELEMENT VALUE
 5167  022704  005203                  18$:    INC     R3              ;INCR ROW ELEMENT VALUE
 5168  022706  020327  000020                  CMP     R3,#20          ;SEE IF ROW ELEMENT IS VALID
 5169  022712  002401                          BLT     20$             ;BR IF VALUE IS OK
 5170  022714  005003                          CLR     R3              ;RESET VALUE TO 0
 5171  022716  020301                  20$:    CMP     R3,R1           ;SEE IF VALUE IS > ACTUAL DRIVES
 5172  022720  002403                          BLT     22$             ;BR IF VALUE OK
 5173  022722  112720  000377                  MOVB    #377,(R0)+      ;SET INVALID ENTRY = 377
 5174  022726  000401                          BR      24$             ;PROCEED
 5175  022730  110320                  22$:    MOVB    R3,(R0)+        ;SET VALUE IN TABLE
```

```
 5176  022732  005304                  24$:    DEC     R4              ;DECR ROW ELEMENT COUNTER
 5177  022734  001363                          BNE     18$             ;BR IF NOT DONE WITH ROW YET
 5178  022736  020027  006512                  CMP     R0,#TABLED+256. ;SEE IF DONE WITH TABLE YET
 5179  022742  103755                          BLO     16$             ;BR IF NOT DONE YET
 5180                                  ;SET INVALID ELEMENTS OF FIRST COLUMN = 377
 5181  022744  012700  006112                  MOV     #TABLED,R0      ;INIT COLUMN POINTER
 5182  022750  012703  000020                  MOV     #16.,R3         ;INIT COL. ELEMENT COUNTER
 5183  022754  121001                  26$:    CMPB    (R0),R1         ;SEE IF ELEMENT IS VALID
 5184  022756  002402                          BLT     30$             ;BR IF YES
 5185  022760  112710  000377                  MOVB    #377,(R0)       ;SET INVALID ELEMENT = 377
 5186  022764  062700  000020          30$:    ADD     #16.,R0         ;INCREMENT COLUMN POINTER
 5187  022770  005303                          DEC     R3              ;SEE IF ALL DONE YET
 5188  022772  001370                          BNE     26$             ;BR IF NOT ALL DONE YET
 5189  022774  104410                          RESREG                  ;RESTORE R0-R5
 5190  022776  000207                          RTS     PC              ;RETURN
 5191
 5192
 5193
 5194                                  ;;*****************************************************************
 5195                                  .SBTTL  ROTBLD - ROTATE TABLED RIGHT 3 COLUMNS
 5196                                  ;*THIS SUBROUTINE ROTATES ALL ELEMENTS OF TABLED RIGHT 3 COLUMNS,
 5197                                  ;*SO THAT THE DRIVE DATA WILL BE ROTATED SEVERAL SECTORS IN EACH
 5198                                  ;*CURRENT ZONE.
 5199                                  ;;*****************************************************************
 5200  023000  104407                  ROTBLD: SAVREG                  ;SAVE R0-R5
 5201  023002  012700  006512                  MOV     #TABLED+256.,R0 ;POINT TO END OF TABLED
 5202  023006  010001                          MOV     R0,R1           ;GET A COPY IN R1
 5203  023010  012704  000015          4$:     MOV     #13.,R4         ;INIT ROW ELEMENT COUNTER
 5204  023014  114037  001304                  MOVB    -(R0),$TMP11    ;SAVE LAST 3 ELEMENTS IN THIS ROW
 5205  023020  114037  001306                  MOVB    -(R0),$TMP12
 5206  023024  114037  001310                  MOVB    -(R0),$TMP13
 5207  023030  114041                  6$:     MOVB    -(R0),-(R1)     ;MOVE AN ELEMENT RIGHT 3 PLACES
 5208  023032  005304                          DEC     R4              ;DECR ROW ELEMENT COUNTER
 5209  023034  001375                          BNE     6$              ;BR IF NOT DONE YET
 5210  023036  113741  001304                  MOVB    $TMP11,-(R1)    ;MOVE IN 3 SAVED ELEMENTS
 5211  023042  113741  001306                  MOVB    $TMP12,-(R1)
 5212  023046  113741  001310                  MOVB    $TMP13,-(R1)
 5213  023052  020027  006112                  CMP     R0,#TABLED      ;SEE IF DONE WITH TABLE YET
 5214  023056  101354                          BHI     4$              ;BR IF NOT DONE YET
 5215  023060  104410                          RESREG                  ;RESTORE R0-R5
 5216  023062  000207                          RTS     PC              ;RETURN
 5217
 5218
 5219
 5220                                  ;;*****************************************************************
 5221                                  .SBTTL  WRTBLK - WRITE ONE CYLINDER BLOCK
 5222                                  ;*THIS SUBROUTINE WRITES ALL THE SECTORS FOR THE CURRENT DRIVE
 5223                                  ;*INTO THE CYLINDER BLOCK WHOSE STARTING CYL NUMBER IS IN
 5224                                  ;*P.CYLN(R5) ON ENTRY. THE DATA IS WRITTEN ON ALL  SURFACES
 5225                                  ;*THE LOGICAL DRIVE NUMBER MUST BE IN LOGDRV ON ENTRY.
 5226                                  ;*
 5227                                  ;*IT CALLS WRTSEC TO ACTUALLY WRITE EACH 256-WORD SECTOR.
 5228                                  ;;*****************************************************************
 5229  023064  104407                  WRTBLK: SAVREG                  ;SAVE R0-R5
 5230  023066  113700  003154                  MOVB    LOGDRV,R0       ;GET LOGICAL DRIVE NO.
 5231  023072  006300                          ASL     R0              ;CONVERT IT TO DATA PATTERN INDEX
```

```
 5232  023074  016003  006512                    MOV     TABLEG(R0),R3   ;GET DATA PATTERN WORD INTO R3
 5233  023100  012701  006112                    MOV     #TABLED,R1      ;GET TABLED POINTER
 5234  023104  012704  000020                    MOV     #16.,R4         ;INIT CYLINDER COUNTER
 5235  023110  010100                    4$:     MOV     R1,R0
 5236  023112  123720  003154            6$:     CMPB    LOGDRV,(R0)+    ;SEE IF DRIVE LISTED HERE
 5237  023116  001375                            BNE     6$              ;BR IF NOT FOUND YET IN THIS ROW
 5238  023120  005300                            DEC     R0              ;BACK UP  BY 1 BYTE
 5239  023122  160100                            SUB     R1,R0           ;COMPUTE ROW POSITION
 5240  023124  116065  006572  000004            MOVB    SECLST(R0),P.SECT(R5) ;GET SECTOR NO. FROM ROW POSITION
 5241  023132  105065  000005                    CLRB    P.TRCK(R5)      ;SET TRACK = 0
 5242  023136  004737  025746            8$:     JSR     PC,WRTSEC       ;WRITE 256 WORDS AT THIS SECTOR
 5243  023142  105265  000005                    INCB    P.TRCK(R5)      ;INCR TRACK NO.
 5244  023146  126527  000005  000002            CMPB    P.TRCK(R5),#2   ;SEE IF ALL TRACKS DONE YET
 5245  023154  003770                            BLE     8$              ;BR IF NOT DONE YET
 5246  023156  062701  000020                    ADD     #16.,R1         ;POINT TO START OF NEXT ROW
 5247  023162  005265  000002                    INC     P.CYLN(R5)      ;INCR CYLINDER NO.
 5248  023166  005304                            DEC     R4              ;DECREMENT CYLINDER COUNTER
 5249  023170  001347                            BNE     4$              ;BR IF 16 CYLS NOT DONE YET
 5250  023172  104410                            RESREG                  ;RESTORE R0-R5
 5251  023174  000207                            RTS     PC              ;RETURN
 5252
 5253
 5254
 5255                                    ;;*****************************************************************
 5256                                    ;*CKSCOR - DECREMENT DRIVE SCORES ROUTINE
 5257                                    ;*THIS SUBROUTINE CHECKS TO SEE IF THE PROGRAM IS CURRENTLY
 5258                                    ;*READING OVRWRT OR COMPAT DATA WITH OFFSETS AND HANDLING DATA CHECK
 5259                                    ;*ERRORS. IF SO, THE SCORE FOR THE DRIVE WHICH WROTE THE
 5260                                    ;*FAILING DATA IS DECREMENTED.
 5261                                    ;*ON ENTRY, THE BYTE LABELED "TSTTYP" HAS THE FOLLOWING
 5262                                    ;*POSSIBLE VALUES :
 5263                                    ;*      TSTTYP = 0  - NOT PRESENTLY READING WITH OFFSET
 5264                                    ;*             = 1  - PRESENTLY READING OVRWRT DATA WITH OFFSET
 5265                                    ;*             = 2  - PRESENTLY READING COMPAT DATA WITH OFFSET
 5266                                    ;*             = 3  - PRESENTLY READING SELF TEST DATA WITH OFFSET
 5267                                    ;;*****************************************************************
 5268  023176  104407                    CKSCOR: SAVREG                  ;SAVE R0-R5
 5269  023200  105737  003143                    TSTB    TSTTYP          ;SEE IF CURRENTLY TESTING WITH OFFSET
 5270  023204  001474                            BEQ     20$             ;BR TO EXIT, IF NOT
 5271  023206  032737  000024  005424            BIT     #DCKERR!HVRCER,RECODE ;SEE IF A DCK OR HVRC ERR OCCURRED
 5272  023214  001470                            BEQ     20$             ;BR IF NOT
 5273                                    ;GET THE NO. OF THE DRIVE WHICH WROTE DATA WHICH CAUSED DCK OR HVRC ERR
 5274  023216  004737  034276                    JSR     PC,GTPKAD       ;COMPUTE PACK ADDRESS OF ERROR
 5275  023222  122737  000003  003143            CMPB    #3,TSTTYP       ;SEE IF DOING SELF-TEST
 5276  023230  001002                            BNE     4$              ;BR IF NOT
 5277  023232  005004                            CLR     R4              ;SET DRIVE NO. = 0
 5278  023234  000431                            BR      10$             ;CONTINUE
 5279  023236  013701  001174            4$:     MOV     $REG5,R1        ;GET CYL NO. OF ERROR
 5280  023242  042701  177760                    BIC     #177760,R1      ;CLEAR HI CYL BITS
 5281  023246  006301                            ASL     R1              ;CONVERT TO TABLED INDEX FOR ROW
 5282  023250  006301                            ASL     R1
 5283  023252  006301                            ASL     R1
 5284  023254  006301                            ASL     R1
 5285  023256  062701  006112                    ADD     #TABLED,R1      ;COMPUTE POINTER TO TABLED ROW
 5286  023262  005000                            CLR     R0              ;INIT SECLST INDEX TO 0
 5287  023264  123760  001200  006572    6$:     CMPB    $REG7,SECLST(R0) ;SEE IF SECTOR IS LISTED IN SECLST
```

```
 5288  023272  001405                        BEQ     8$              ;BR IF YES
 5289  023274  005200                        INC     R0              ;INCR SECLST INDEX
 5290  023276  020027  000020                CMP     R0,#16.         ;SEE IF WHOLE LIST CHECKED YET
 5291  023302  002770                        BLT     6$              ;BR IF NOT YET
 5292  023304  000434                        BR      20$             ;SECTOR NOT LISTED - EXIT, BECAUSE
 5293                                                                ;  FAILING DRIVE IS NOT BEING TESTED
 5294  023306  060001                8$:     ADD     R0,R1           ;COMPUTE TABLED POINTER TO FAILING DRIVE
 5295  023310  111104                        MOVB    (R1),R4         ;GET ACTUAL DRIVE NO.
 5296  023312  122704  000377                CMPB    #377,R4         ;SEE IF DRIVE IS NOT BEING TESTED
 5297  023316  001427                        BEQ     20$             ;BR IF DRIVE NOT TESTED
 5298  023320  012700  007432        10$:    MOV     #NSCOR0,R0      ;GET ADRS OF NEG SCRATCH SCORES
 5299  023324  105737  003156                TSTB    OFSDIR          ;SEE WHICH OFST DIRECTION
 5300  023330  001402                        BEQ     12$             ;BR IF NEG
 5301  023332  012700  007512                MOV     #PSCOR0,R0      ;GET ADRS OF POS SCRATCH SCORES
 5302  023336  105737  001176        12$:    TSTB    $REG6           ;SEE IF TRACK = 0
 5303  023342  001410                        BEQ     14$             ;BR IF YES
 5304  023344  062700  000020                ADD     #16.,R0         ;POINT TO SCORES FOR TRACK 1
 5305  023350  123727  001176  000001        CMPB    $REG6,#1        ;SEE IF TRACK = 1
 5306  023356  001402                        BEQ     14$             ;BR IF YES
 5307  023360  062700  000020                ADD     #16.,R0         ;POINT TO SCORES FOR TRACK 2
 5308  023364  060400                14$:    ADD     R4,R0           ;COMPUTE POINTER TO SCORE FOR DRIVE
 5309  023366  121037  003157                CMPB    (R0),OFSCNT     ;SEE IF THIS SCORE IS PERFECT SO FAR
 5310  023372  001001                        BNE     20$             ;EXIT IF NOT
 5311  023374  105310                        DECB    (R0)            ;DECREMENT SCORE FOR THIS DRIVE
 5312  023376  104410                20$:    RESREG                  ;RESTORE R0-R5
 5313  023400  000207                        RTS     PC              ;RETURN
 5314
 5315
 5316
 5317                                ;;*****************************************************************
 5318                                .SBTTL  RDCHRS - READ A STRING OF KBD INPUT CHARS
 5319                                ;*THIS SUBROUTINE READS A STRING OF UP TO EIGHTY INPUT
 5320                                ;*CHARACTERS AT THE KBD, TERMINATED BY <CR>, AND PLACES
 5321                                ;*THEM IN BUFFO, TERMINATED BY A NULL (0) BYTE.
 5322                                ;*RUB-OUT AND (^U) FEATURES ARE PROVIDED.
 5323                                ;*IMMEDIATELY FOLLOWING THE SUBROUTINE CALL, THREE SPECIAL
 5324                                ;*RETURN ADDRESSES MUST BE LISTED: THE FIRST RETURN IS
 5325                                ;*TAKEN IF (^C) IS TYPED, THE SECOND IS TAKEN IF (^Z) IS
 5326                                ;*TYPED, AND THE THIRD IS TAKEN IF (^U) OR INVALID INPUT IS TYPED.
 5327                                ;*IF (^G) IS TYPED, THE SOFTWARE SWITCH REGISTER IS OPENED
 5328                                ;*FOR MODIFICATION, IF SELECTED, AND THEN THE (^G) RETURN
 5329                                ;*IS TAKEN.
 5330                                ;*THE NUMBER OF INPUT CHARACTERS IN THE BUFFER IS RETURNED
 5331                                ;*IN R0.
 5332                                ;*
 5333                                ;*  CALL -      JSR     PC,RDCHRS
 5334                                ;*              <CONTROL-C RETURN ADDRESS>
 5335                                ;*              <CONTROL-Z RETURN ADDRESS>
 5336                                ;*              <CONTROL-U OR ERROR RETURN ADDRESS>
 5337                                ;*              RETURN
 5338                                ;;*****************************************************************
 5339
 5340  023402                        RDCHRS:
 5341  023402  010146                        MOV     R1,-(SP)        ;SAVE R1
 5342  023404  010246                        MOV     R2,-(SP)        ;SAVE R2
 5343  023406  005000                        CLR     R0              ;INITIALIZE CHARACTER COUNT
```

```
5344  023410  005001                          CLR     R1              ;INITIALIZE RUB-OUT INDICATOR
5345                                  ;READ A CHARACTER
5346  023412  104406                  2$:     RDCHR                   ;READ A CHARACTER
5347  023414  112602                          MOVB    (SP)+,R2        ;GET CHARACTER INTO R2
5348                                  ;CHECK FOR (^C)
5349  023416  122702  000003                  CMPB    #003,R2         ;SEE IF (^C) TYPED
5350  023422  001006                          BNE     4$              ;BR IF NOT (^C)
5351  023424  104401  012052                  TYPE    ,CNTRLC         ;ECHO (^C)
5352  023430  017666  000004  000004  3$:     MOV     @4(SP),4(SP)    ;PUT RETURN ADDRESS ON STACK
5353  023436  000523                          BR      24$             ;BR TO TAKE EXIT
5354                                  ;CHECK FOR (^Z)
5355  023440  122702  000032          4$:     CMPB    #032,R2         ;SEE IF (^Z) TYPED
5356  023444  001006                          BNE     6$              ;BR IF NOT (^Z)
5357  023446  104401  012057                  TYPE    ,CNTRLZ         ;ECHO (^Z)
5358  023452  062766  000002  000004          ADD     #2,4(SP)        ;MAKE OLD PC POINT TO NEXT RETURN ADR.
5359  023460  000763                          BR      3$              ;BR TO TAKE (^Z) EXIT
5360                                  ;CHECK FOR (^U)
5361  023462  122702  000025          6$:     CMPB    #025,R2         ;SEE IF (^U) TYPED
5362  023466  001006                          BNE     8$              ;BR IF NOT (^U)
5363  023470  104401  012071                  TYPE    ,CNTRLU         ;ECHO (^U)
5364  023474  062766  000004  000004  7$:     ADD     #4,4(SP)        ;MAKE OLD PC POINT TO NEXT RETURN ADDR.
5365  023502  000752                          BR      3$              ;BR TO TAKE (^U) EXIT
5366                                  ;CHECK FOR (^G)
5367  023504  122702  000007          8$:     CMPB    #007,R2         ;SEE IF (^G) TYPED
5368  023510  001005                          BNE     9$              ;BR IF NOT (^G)
5369  023512  104401  012076                  TYPE    ,CNTRLG         ;ECHO (^G)
5370  023516  004737  021106                  JSR     PC,GTSWRG       ;OPEN SOFTWARE SWITCH REG. FOR CHANGE
5371  023522  000764                          BR      7$              ;TAKE (^U) RETURN
5372                                  ;CHECK FOR RUB-OUT (DELETE)
5373  023524  122702  000177          9$:     CMPB    #177,R2         ;SEE IF RUB-OUT (DEL) TYPED
5374  023530  001020                          BNE     14$             ;BR IF NOT RUB-OUT
5375  023532  005700                          TST     R0              ;CHECK THE CHARACTER COUNT
5376  023534  001726                          BEQ     2$              ;BR IF COUNT = 0
5377  023536  005701                          TST     R1              ;CHECK THE RUB-OUT INDICATOR
5378  023540  001003                          BNE     11$             ;BR IF WE HAD A PREVIOUS RUB-OUT
5379  023542  005201                          INC     R1              ;SET RUB-OUT INDICATOR
5380  023544  104401  012107                  TYPE    ,BKSLSH         ;TYPE A BACK-SLASH (\)
5381  023550  005037  005444          11$:    CLR     SCRACH          ;USE SCRATCH WORD FOR TEMP. BUFFER
5382  023554  005300                          DEC     R0              ;DECREMENT COUNT
5383  023556  116037  005212  005444          MOVB    BUFFO(R0),SCRACH ;GET LAST CHAR. INTO BUFFER
5384  023564  104401  005444                  TYPE    ,SCRACH         ;ECHO CHARACTER TO BE DELETED
5385  023570  000710                          BR      2$              ;GO READ ANOTHER CHARACTER
5386  023572  005701                  14$:    TST     R1              ;CHECK THE RUB-OUT INDICATOR
5387  023574  001403                          BEQ     16$             ;BR IF INDICATOR IS NOT SET
5388  023576  104401  012107                  TYPE    ,BKSLSH         ;TYPE A BACK-SLASH
5389  023602  005001                          CLR     R1              ;CLEAR THE RUB-OUT INDICATOR
5390                                  ;CHECK FOR CARRIAGE RETURN
5391  023604  122702  000015          16$:    CMPB    #015,R2         ;SEE IF <CR> TYPED
5392  023610  001426                          BEQ     19$             ;BR IF <CR>
5393                                  ;HANDLE POSSIBLE DIGIT
5394  023612  005037  005444                  CLR     SCRACH          ;USE SCRATCH WORD FOR TEMP. BUFFER
5395  023616  110237  005444                  MOVB    R2,SCRACH       ;GET THIS CHARACTER INTO BUFFER
5396  023622  104401  005444                  TYPE    ,SCRACH         ;ECHO THE CHARACTER TYPED
5397  023626  110260  005212                  MOVB    R2,BUFFO(R0)    ;PUT CHARACTER INTO BUFFER
5398  023632  005200                          INC     R0              ;INCREMENT CHARACTER COUNTER
5399  023634  022700  000120                  CMP     #80.,R0         ;SEE IF TOO MANY CHARACTERS TYPED
```

```
5400  023640  001264                        BNE     2$                ;BR IF NOT TOO MANY
5401  023642  104401  001325                TYPE    ,$CRLF            ;TYPE <CR> AND <LF>
5402  023646  112760  000000  005212        MOVB    #0,BUFFO(R0)  ;PUT TERMINATING NULL INTO BUFFER
5403  023654  104401  005212                TYPE    ,BUFFO            ;ECHO INPUT STRING
5404  023660  104401  001324                TYPE    ,$QUES            ;TYPE <?>,<CR>,<LF>
5405  023664  000703                        BR      7$                ;TAKE ERROR EXIT FROM RDCHRS
5406  023666  104401  001325        19$:    TYPE    ,$CRLF            ;TYPE <CR>,<LF>
5407  023672  112760  000000  005212        MOVB    #0,BUFFO(R0)      ;PUT TERMINATING NULL INTO BUFFER
5408  023700  062766  000006  000004        ADD     #6,4(SP)          ;FIX UP RETURN ADDRESS
5409  023706  012602                24$:    MOV     (SP)+,R2          ;RESTORE R2
5410  023710  012601                        MOV     (SP)+,R1          ;RESTORE R1
5411  023712  000207                        RTS     PC                ;SUBROUTINE EXIT
5412
5413
5414
5415                                ;;*****************************************************************
5416                                .SBTTL  DRVSER - TYPE DRIVE SERIAL NUMBER (LOW 3 DIGITS)
5417                                ;*THIS SUBROUTINE TYPES "DRIVE SER. NO.  XXX" (IN DECIMAL), WITH LEADING
5418                                ;*ZEROS SUPPRESSED.
5419                                ;;*****************************************************************
5420  023714  104407                DRVSER: SAVREG                    ;SAVE R0-R5
5421  023716  004737  021140                JSR     PC,INITSS         ;CLEAR S.S. AND PARAMETERS
5422  023722  112765  000141  000001        MOVB    #RDSTAT,P.CMND(R5) ;SET READ STATUS COMMAND
5423  023730  004737  030124                JSR     PC,DRVCAL         ;READ STATUS OF THIS DRIVE
5424  023734  104401  011701                TYPE    ,DRIV             ;TYPE "DRIVE"
5425  023740  104401  011717                TYPE    ,SERNM            ;TYPE " SER. NO.  "
5426  023744  016501  000054                MOV     P.A11(R5),R1      ;GET "A" STATUS BYTE 11
5427  023750  012704  043144                MOV     #$OCTVL,R4        ;GET ADDR OF CHAR BUFFER
5428  023754  010446                        MOV     R4,-(SP)          ;STORE IT ON STACK FOR $SUPRS
5429  023756  012703  000003                MOV     #3,R3             ;INIT CHAR COUNT
5430  023762  006101                        ROL     R1                ;INITIALIZE BIT POSITIONS
5431  023764  006101                        ROL     R1
5432  023766  006101                4$:     ROL     R1                ;GET NEXT 4 BITS
5433  023770  006101                        ROL     R1
5434  023772  006101                        ROL     R1
5435  023774  006101                        ROL     R1
5436  023776  010100                        MOV     R1,R0             ;GET A WORKING COPY
5437  024000  042700  177760                BIC     #177760,R0        ;CLEAR ALL BUT LOW 4 BITS
5438  024004  052700  000060                BIS     #'0,R0            ;CONVERT A DIGIT TO ASCII
5439  024010  110024                        MOVB    R0,(R4)+          ;PUT ASCII DIGIT INTO CHAR BUFFER
5440  024012  005303                        DEC     R3                ;DECREMENT CHAR COUNT
5441  024014  001364                        BNE     4$                ;BR IF NOT 3 CHARS YET
5442  024016  105014                        CLRB    (R4)              ;INSERT NULL TERMINATOR
5443  024020  004737  043356                JSR     PC,@#$SUPRS       ;TYPE DRIVE SER. NUMBER
5444  024024  104401  001325                TYPE    ,$CRLF            ;TYPE <CR> AND <LF>
5445  024030  104410                        RESREG                    ;RESTORE R0-R5
5446  024032  00C207                        RTS     PC                ;RETURN
5447
5448
5449
5450                                ;;*****************************************************************
5451                                .SBTTL  CRTSER - TYPE CARTRIDGE SERIAL NUMBER
5452                                ;*THIS SUBROUTINE TYPES "CART. SER. NO.  XXXXXXXXXX" (IN OCTAL),
5453                                ;*WITH LEADING ZEROS SUPPRESSED.
5454                                ;;*****************************************************************
5455  024034  004737  021140        CRTSER: JSR     PC,INITSS         ;CLEAR S.S. AND PARAMETERS
```

```
 5456  024040  142765  000020  000007          BICB    #B.CFMT,P.CS1H(R5)      ;SET 22 SECTOR FORMAT
 5457  024046  105737  003134                  TSTB    FORMAT          ;CHECK THE ACTUAL FORMAT
 5458  024052  001402                          BEQ     4$              ;BR IF 22 SECTORS
 5459  024054  105265  000004                  INCB    P.SECT(R5)      ;IF 20 SECTORS, READ SECTOR 1
 5460  024060  112765  000121  000001  4$:     MOVB    #RDDATA,P.CMND(R5) ;SET READ COMMAND
 5461  024066  013765  024362  000002          MOV     LSTCYL,P.CYLN(R5) ;SET CYL = 632(OCT)/1456(OCT)
 5462  024074  112765  000002  000005          MOVB    #LSTTRK,P.TRCK(R5) ;SET TRACK = 2
 5463  024102  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;SET READ BUFFER ADDRESS
 5464  024110  012765  177776  000012          MOV     #-2,P.WC(R5)    ;SET WORD COUNT TO READ 2 WORDS
 5465  024116  004737  030124                  JSR     PC,DRVCAL       ;READ SERIAL NO. IN BSF
 5466  024122  104401  011710                  TYPE    ,CART           ;TYPE  "CART."
 5467  024126  104401  011717                  TYPE    ,SERNM          ;TYPE "SER. NO.  "
 5468  024132  012746  053330                  MOV     #RWBUF,-(SP)    ;GET POINTER FOR $DB2O
 5469  024136  004737  043042                  JSR     PC,@#$DB2O      ;CONVERT BINARY TO OCTAL
 5470  024142  004737  043356                  JSR     PC,@#$SUPRS     ;TYPE CART. SERIAL NO. IN OCTAL
 5471  024146  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR> AND <LF>
 5472  024152  000207                          RTS     PC              ;RETURN
 5473
 5474                                  ;THIS ROUTINE ASKS THE OPERATOR FOR THE DRIVE TYOPE AND ALL FUTURE
 5475                                  ;TESTS WILL BE PERFORMED ASSUMING ALL THE DRIVES ARE OF THAT TYPE
 5476
 5477  024154  104401  001325          GETTYP: TYPE    ,$CRLF
 5478  024160  104401  007775                  TYPE    ,ASKTYP         ;ASK DRIVE TYPE
 5479  024164  004737  023402                  JSR     PC,RDCHRS       ;READ INPUT STRING
 5480  024170  024154                          GETTYP                  ;CONT-C RETURN
 5481  024172  024154                          GETTYP                  ;CONT-Z RET
 5482  024174  024154                          GETTYP                  ;CONT-U RET
 5483  024176  005700                          TST     R0              ;SEE IF CHAR TYPED
 5484  024200  001005                          BNE     2$              ;BR IF NO
 5485  024202  104401  005212          1$:     TYPE    ,BUFFO          ;ECHO BAD INPUT
 5486  024206  104401  001324                  TYPE    ,$QUES          ;TYPE <?>
 5487  024212  000760                          BR      GETTYP          ;ASK AGAIN
 5488
 5489  024214  022737  000066  005212  2$:     CMP     #'6,BUFFO       ;SEE IF 6 TYPED
 5490  024222  001025                          BNE     3$              ;BR IF NO
 5491  024224  105037  003133                  CLRB    TYPFMT          ;SETUP FOR RK06
 5492  024230  012737  000632  024362          MOV     #632,LSTCYL
 5493  024236  012737  000025  024364          MOV     #25,HOLD1
 5494  024244  012737  010025  024366          MOV     #10025,HOLD2
 5495  024252  012737  000100  024370          MOV     #100,HOLD3
 5496  024260  012737  000060  024372          MOV     #60,HOLD4
 5497  024266  012737  000040  024374          MOV     #40,HOLD5
 5498  024274  000207                          RTS     PC
 5499  024276  022737  000067  005212  3$:     CMP     #'7,BUFFO       ;SEE IF 7 TYPED
 5500  024304  001336                          BNE     1$              ;BR IF NO
 5501  024306  152737  000004  003133          BISB    #B.CDT,TYPFMT   ;SETUP FOR RK07
 5502  024314  012737  001456  024362          MOV     #1456,LSTCYL
 5503  024322  012737  002025  024364          MOV     #2025,HOLD1
 5504  024330  012737  012025  024366          MOV     #12025,HOLD2
 5505  024336  012737  000200  024370          MOV     #200,HOLD3
 5506  024344  012737  000160  024372          MOV     #160,HOLD4
 5507  024352  012737  000100  024374          MOV     #100,HOLD5
 5508  024360  000207                          RTS     PC
 5509
 5510  024362  000000                  LSTCYL: 0
 5511  024364  000000                  HOLD1:  0
```

```
5512  024366  000000                  HOLD2:  0
5513  024370  000000                  HOLD3:  0                       ;CYL DIF
5514  024372  000000                  HOLD4:  0                       ;COMPAT START CYL
5515  024374  000000                  HOLD5:  0                       ;HALF CYL DIF
5516
5517
5518
5519
5520
5521
5522
5523                                  ;;*****************************************************************
5524                                  .SBTTL  STALL - STALL FOR ST UNIT STALL TIMES
5525                                  ;*IF SWR BIT 8 = 0, THIS SUBROUTINE STALLS FOR ST STALL TIMES,
5526                                  ;*WHERE A STALL TIME = 40 US, FOR AN "AVERAGE" CPU. IF BIT 8
5527                                  ;*IS EQUAL TO 1, A RANDOM STALL IS APPLIED.
5528                                  ;* CALL -   JSR     PC,STALL
5529                                  ;;*****************************************************************
5530
5531  024376  010046                  STALL:  MOV     R0,-(SP)        ;SAVE R0
5532  024400  010146                          MOV     R1,-(SP)        ;SAVE R1
5533  024402  032777  000400  154530          BIT     #BIT08,@SWR     ;APPLY RANDOM STALL ?
5534  024410  001407                          BEQ     1$              ;BR IF NOT RANDOM
5535  024412  004737  042240                  JSR     PC,$RAND        ;GENERATE PSEUDO-RANDOM NUMBER
5536  024416  013700  042340                  MOV     $LONUM,R0       ;GET IT INTO R0
5537  024422  006200                          ASR     R0              ;SCALE IT DOWN
5538  024424  006200                          ASR     R0
5539  024426  000403                          BR      2$              ;GO STALL WITH RANDOM NO.
5540  024430  013700  005432          1$:     MOV     STALLS,R0       ;GET REQUESTED NO. OF STALLS
5541  024434  001406                          BEQ     6$              ;RETURN IF NO STALL REQUIRED
5542  024436  012701  000016          2$:     MOV     #14.,R1         ;SET CONSTANT FOR 40 US
5543  024442  005301                  4$:     DEC     R1              ;INNER LOOP COUNTER
5544  024444  001376                          BNE     4$              ;INNER LOOP BR
5545  024446  005300                          DEC     R0              ;OUTER LOOP COUNTER
5546  024450  001372                          BNE     2$              ;OUTER LOOP BR
5547  024452  012601                  6$:     MOV     (SP)+,R1        ;RESTORE R1
5548  024454  012600                          MOV     (SP)+,R0        ;RESTORE R0
5549  024456  000207                          RTS     PC              ;RETURN
5550
5551
5552
5553                                  ;;*****************************************************************
5554                                  ;* LODSEC - THIS SUBROUTINE LOADS THE CONTENTS OF THE DATA PATTERN
5555                                  ;*IN TABLEB INTO ALL 256(DEC) WORDS OF THE DATA BUFFER (RWBUF).
5556                                  ;;*****************************************************************
5557  024460  104407                  LODSEC: SAVREG                  ;SAVE R0-R5
5558  024462  012701  053330                  MOV     #RWBUF,R1       ;GET ADRS OF R/W BUF INTO R1
5559  024466  012700  005766          2$:     MOV     #TABLEB,R0      ;GET DATA PATTERN ADDRESS
5560  024472  012703  000020                  MOV     #16.,R3         ;INIT PATTERN COUNTER
5561  024476  012021                  4$:     MOV     (R0)+,(R1)+     ;MOVE A PATTERN WORD
5562  024500  020127  054330                  CMP     R1,#RWBUF+512.  ;SEE IF DONE YET
5563  024504  103003                          BHIS    9$              ;BR IF DONE
5564  024506  005303                          DEC     R3              ;DECREMENT PATTERN COUNTER
5565  024510  001372                          BNE     4$              ;BR IF NOT RESTARTING PATTERN YET
5566  024512  000765                          BR      2$              ;BR TO RESTART PATTERN
5567  024514  104410                  9$:     RESREG                  ;RESTORE R0-R5
```

```
 5568  024516  000207                          RTS     PC              ;RETURN
 5569
 5570
 5571
 5572
 5573
 5574
 5575                                  ;;*****************************************************************
 5576                                  .SBTTL  FINADR - COMPUTE FINAL PACK ADDRESS
 5577                                  ;*THIS SUBROUTINE IS USED AFTER A DATA TRANSFER HAS COMPLETED, TO
 5578                                  ;*COMPUTE THE FINAL PACK ADDRESS AT TERMINATION. THE 2'S COMP. OF THE
 5579                                  ;*WORD NO. AT THE POINT OF INTEREST IS PASSED IN LASTWC. THIS IS USED WITH
 5580                                  ;*P.CYLN(R5), P.TRCK(R5), AND P.SECT(R5) TO COMPUTE THE CORRESPONDING
 5581                                  ;*PACK ADDRESS, WHICH IS RETURNED IN FINCYL, FINTRK, AND FINSEC.
 5582                                  ;*THE SUBROUTINE IS USED TO DETERMINE THE PACK ADDRESS OF A SOFTWARE
 5583                                  ;*DATA MISCOMPARE.
 5584                                  ;;*****************************************************************
 5585  024520  104407                  FINADR: SAVREG                  ;SAVE R0-R5
 5586  024522  013746  005474                  MOV     LASTWC,-(SP)    ;STORE WORD COUNT
 5587  024526  005416                          NEG     (SP)            ;MAKE IT POSITIVE
 5588  024530  005046                  18$:    CLR     -(SP)           ;MAKE ROOM ON STACK
 5589  024532  116616  000003                 MOVB    3(SP),(SP)      ;STORE NO. OF SECTORS TRANSFERRED
 5590  024536  005066  000002                 CLR     2(SP)           ;CLEAR LOCATION ON STACK
 5591  024542  116566  000004  000002         MOVB    P.SECT(R5),2(SP) ;STORE STARTING SECTOR
 5592  024550  066616  000002                 ADD     2(SP),(SP)      ;DETERMINE FINAL SECTOR ADDRESS
 5593  024554  005066  000002                 CLR     2(SP)           ;CLEAR NO. OF TRACKS TRANSFERRED
 5594  024560  012700  000026                 MOV     #22.,R0         ;SET FOR 22 SECTORS
 5595  024564  105737  003134                 TSTB    FORMAT          ;DETERMINE THE FORMAT
 5596  024570  001402                          BEQ     19$             ;BR IF 22 SECTORS
 5597  024572  012700  000024                 MOV     #20.,R0         ;SET FOR 20 SECTORS
 5598  024576  020016                  19$:    CMP     R0,(SP)         ;CHECK FOR SECTOR OVERFLOW
 5599  024600  101004                          BHI     20$             ;NO, CHECK IF SECTOR CORRECT
 5600  024602  160016                          SUB     R0,(SP)         ;DECREMENT SECTOR COUNT BY 20 OR 22
 5601  024604  005266  000002                 INC     2(SP)           ;INCREMENT TRACKS TRANSFERRED
 5602  024610  000772                          BR      19$             ;CHECK FOR SECTOR OVERFLOW
 5603  024612  112637  005473          20$:    MOVB    (SP)+,FINSEC    ;STORE FINAL SECTOR
 5604  024616  005046                          CLR     -(SP)           ;MAKE ROOM FOR TRACKS TRANSFERRED
 5605  024620  116516  000005                 MOVB    P.TRCK(R5),(SP) ;STORE STARTING TRACK
 5606  024624  066616  000002                 ADD     2(SP),(SP)      ;DETERMINE FINAL TRACK ADDRESS
 5607  024630  005066  000002                 CLR     2(SP)           ;CLEAR FINAL CYLINDER
 5608  024634  122716  000003          21$:    CMPB    #3,(SP)         ;CHECK FOR TRACK OVERFLOW
 5609  024640  101005                          BHI     22$             ;NO, CHECK FINAL TRACK
 5610  024642  162716  000003                 SUB     #3,(SP)         ;DECREMENT TRACK COUNT BY 3
 5611  024646  005266  000002                 INC     2(SP)           ;INCR CYL COUNT
 5612  024652  000770                          BR      21$             ;CHECK FOR TRACK OVERFLOW
 5613  024654  112637  005472          22$:    MOVB    (SP)+,FINTRK    ;STORE FINAL TRACK
 5614  024660  066516  000002                 ADD     P.CYLN(R5),(SP) ;CALCULATE FINAL CYLINDER
 5615  024664  011637  005470                 MOV     (SP),FINCYL     ;STORE FINAL  CYLINDER
 5616  024670  005726                          TST     (SP)+           ;CLEAN OFF STACK
 5617  024672  104410                          RESREG                  ;RESTORE R0-R5
 5618  024674  000207                          RTS     PC              ;RETURN
 5619
 5620
 5621
 5622                                  ;;*****************************************************************
 5623                                  .SBTTL  LODBUF - LOAD THE READ/WRITE DATA BUFFER
```

LODBUF - LOAD THE READ/WRITE DATA BUFFER SEQ 0108

```
                                        ;*THIS SUBROUTINE LOADS THE READ/WRITE DATA BUFFER (POINTED TO BY
                                        ;* PMA AND PMA+2) WITH THE APPROPRIATE REPEATING DATA PATTERN. IF
                                        ;*PARAMETER PATRN = 0 ON ENTRY, THE DATA WHICH IS LOADED IS COMPRISED
                                        ;*OF ALL THE PATTERNS 00-15 (QUICK VERIFY DEFAULT DATA TEST).
                                        ;*IF PATRN IS NOT 0, THE NO. OF WORDS IN THE WORD WDSXFR
                                        ;*OF THE PATTERN IDENTIFIED BY THE NO. OF THE BIT SET IN R1 ON
                                        ;*ENTRY, ARE LOADED INTO THE BUFFER.
                                        ;;*****************************************************************

024676  104407                  LODBUF: SAVREG                  ;SAVE R0-R5
024700  013702  005466                  MOV     WDSXFR,R2       ;GET NO. OF WORDS
024704  005737  005446                  TST     PATRN           ;SEE IF QUICK VERIFY DATA TEST DESIRED
024710  001007                          BNE     3$              ;BR IF NOT QUICK VERIFY
024712  012700  005566                  MOV     #PAT00,R0       ;SET DATA PATTERN STARTING ADDRESS
024716  012746  000400                  MOV     #256.,-(SP)     ;SET PATTERN WORD COUNT
024722  012701  053330                  MOV     #RWBUF,R1       ;SET BUFFER ADDRESS
024726  000422                          BR      30$             ;PROCEED
024730  005000                  3$:     CLR     R0              ;INIT PATTERN NUMBER
024732  032701  000001          4$:     BIT     #BIT0,R1        ;SEE IF THIS BIT IS SET
024736  001003                          BNE     6$              ;BR IF THIS BIT IS SET
024740  005200                          INC     R0              ;INCREMENT PATTERN NO.
024742  006201                          ASR     R1              ;SHIFT TO EXAMINE NEXT BIT
024744  000772                          BR      4$              ;BR TO CHECK NEXT BIT
024746  006300                  6$.     ASL     R0              ;MULTIPLY PATTERN NO. BY 32(DEC)
024750  006300                          ASL     R0
024752  006300                          ASL     R0
024754  006300                          ASL     R0
024756  006300                          ASL     R0
024760  062700  005566                  ADD     #PAT00,R0       ;GET ADDRESS OF DESIRED PATTERN
024764  012746  000020                  MOV     #16.,-(SP)      ;SET PATTERN WORD COUNT
024770  013701  005504                  MOV     PMA,R1          ;SET BUFFER ADDRESS
024774  010003                  30$:    MOV     R0,R3           ;GET A COPY OF PATTERN ADDRESS
024776  011604                          MOV     (SP),R4         ;INIT PATTERN WORD COUNT
025000  012321                  34$:    MOV     (R3)+,(R1)+     ;LOAD A DATA WORD INTO BUFFER
025002  005302                          DEC     R2              ;DECREMENT WORD COUNTER
025004  001403                          BEQ     44$             ;BR IF ALL DONE
025006  005304                          DEC     R4              ;DECREMENT PATTERN WORD COUNT
025010  001373                          BNE     34$             ;BR IF NOT DONE WITH PATTERN YET
025012  000770                          BR      30$             ;BR TO REPEAT THE PATTERN
025014  005726                  44$:    TST     (SP)+           ;POP THE STACK
025016  104410                          RESREG                  ;RESTORE R0-R5
025020  000207                          RTS     PC              ;RETURN


                                        ;;*****************************************************************
                                        .SBTTL  CMPBUF - SOFTWARE COMPARE DATA
                                        ;*THIS SUBROUTINE PERFORMS A SOFTWARE COMPARE OF THE R/W DATA BUFFER (POINTED
                                        ;*TO BY PMA AND PMA+2) TO THE APPROPRIATE REPEATING DATA PATTERN. IF
                                        ;*PARAMETER PATRN = 0 ON ENTRY, THE DATA IS COMPRISED OF ALL THE PATTERNS
                                        ;*00-15 (QUICK VERIFY DEFAULT DAT' TEST).
                                        ;*IF PATRN IS NOT 0, THE NO. OF THE BIT SET IN R1 ON ENTRY IDENTIFIES THE
                                        ;*REPEATING DATA PATTERN TO COMPARE AGAINST.
                                        ;;*****************************************************************
025022  104407                  CMPBUF: SAVREG                  ;SAVE R0-R5
025024  112737  000040  051653          MOVB    #40,DH701+38.   ;RESTORE ERROR MSG PARAMS
```

```
 5680  025032  012737  000007  053222          MOV     #7,DF25+2
 5681  025040  013702  005466                  MOV     WDSXFR,R2       ;SET NO. OF WORDS
 5682  025044  005037  005444                  CLR     SCRACH          ;CLEAR COMPARE ERROR COUNT
 5683  025050  004737  031212                  JSR     PC,REPSUP       ;STORE PREv CMND FOR POSS. PRINT
 5684  025054  005737  005446                  TST     PATRN           ;SEE IF THIS IS QUICK VERIFY DEFAULT DATA TEST
 5685  025060  001007                          BNE     3$              ;BR IF NOT
 5686  025062  012700  005566                  MOV     #PAT00,R0       ;GET DATA PATTERN STARTING ADDR
 5687  025066  012746  000400                  MOV     #256.,-(SP)     ;SET PATTERN WORD COUNT
 5688  025072  012701  053330                  MOV     *RWBUF,R1       ;SET BUFFER ADDRESS
 5689  025076  000422                          BR      30$             ;PROCEED
 5690  025100  005000                  3$:     CLR     R0              ;INIT PATTERN NO.
 5691  025102  032701  000001          4$:     BIT     #BIT0,R1        ;SEE IF THIS BIT IS SET
 5692  025106  001003                          BNE     6$              ;BR IF THIS BIT IS SET
 5693  025110  005200                          INC     R0              ;INCREMENT PATTERN NUMBER
 5694  025112  006201                          ASR     R1              ;SHIFT TO EXAMINE NEXT BIT
 5695  025114  000772                          BR      4$              ;BR TO CHECK NEXT BIT
 5696  025116  006300                  6$:     ASL     R0              ;MULTIPLY PATTERN NO. BY 32(DEC)
 5697  025120  006300                          ASL     R0
 5698  025122  006300                          ASL     R0
 5699  025124  006300                          ASL     R0
 5700  025126  006300                          ASL     R0
 5701  025130  062700  005566                  ADD     #PAT00,R0       ;GET ADDRESS OF DESIRED PATTERN
 5702  025134  012746  000020                  MOV     #16.,-(SP)      ;PATTERN WORD COUNT
 5703  025140  013701  005504                  MOV     PMA,R1          ;SET BUFFER ADDRESS
 5704  025144  010003                  30$:    MOV     R0,R3           ;GET COPY OF PATTERN ADDRESS
 5705  025146  011604                          MOV     (SP),R4         ;INIT PATTERN WORD COUNT
 5706  025150  022321                  34$:    CMP     (R3)+,(R1)+     ;COMPARE DATA WORD TO PATTERN WORD
 5707  025152  001002                          BNE     31$             ;BR IF COMPARE ERROR
 5708  025154  000137  025514                  JMP     40$             ;JUMP IF DATA COMPARES OK
 5709  025160  105037  051653          31$:    CLRB    DH701+38.       ;ADJUST DATA HEADER FOR MSG
 5710  025164  012737  000005  053222          MOV     #5,DF25+2       ;ADJUST ERROR DATA WORD COUNT
 5711                          ;COMMON COMPARE ERROR HANDLER
 5712  025172  010237  001202          35$:    MOV     R2,$REG10       ;GET WORD NO.
 5713  025176  163737  005466  001202          SUB     WDSXFR,$REG10
 5714  025204  013737  001202  005474          MOV     $REG10,LASTWC   ;GET 2'S COMP OF WORD NO.
 5715  025212  004737  024520                  JSR     PC,FINADR       ;COMPUTE ACTUAL PACK ADRS
 5716  025216  104407                          SAVREG                  ;SAVE R0-R5
 5717  025220  013700  005470                  MOV     FINCYL,R0       ;GET CYL
 5718  025224  113701  005472                  MOVB    FINTRK,R1       ;GET TRACK
 5719  025230  113702  005473                  MOVB    FINSEC,R2       ;GET SECTOR
 5720  025234  004737  030020                  JSR     PC,BDSRCK       ;SEE IF THIS SECTOR LISTED BAD
 5721  025240  104410                          RESREG                  ;RESTORE R0-R5
 5722  025242  032737  001000  005424          BIT     #BADSEC,RECODE
 5723  025250  001116                          BNE     50$             ;BR IF LISTED- DON'T REPORT ERROR
 5724  025252  005737  005444                  TST     SCRACH          ;CHECK THE ERROR COUNT
 5725  025256  001010                          BNE     36$             ;BR IF THIS IS NOT FIRST ERROR
 5726  025260  104034                  46$:    ERROR   34              ;TYPE HEADING FOR ERROR MSG
 5727  025262  012737  177777  001174  48$:    MOV     #-1,$REG5       ;INIT CYL NO.
 5728  025270  005037  001176                  CLR     $REG6
 5729  025274  005037  001200                  CLR     $REG7
 5730  025300  005237  005444          36$:    INC     SCRACH          ;INCREMENT THE ERROR COUNT
 5731  025304  032777  000001  153626          BIT     #BIT0,@SWR      ;SEE IF ALL ERRORS SHOULD BE REPORTED
 5732  025312  001004                          BNE     38$             ;BR TO REPORT ALL ERRORS
 5733  025314  022737  000012  005444          CMP     #10.,SCRACH     ;SEE IF 10(DEC) ERRORS YET
 5734  025322  002504                          BLT     54$             ;BR IF ERROR LIMIT EXCEEDED
 5735  025324  023737  001174  005470  38$:    CMP     $REG5,FINCYL    ;SEE IF DIFFERENT CYL
```

```
 5736  025332  001010                          BNE     42$             ;BR IF YES
 5737  025334  123737  001176  005472          CMPB    $REG6,FINTRK    ;SEE IF DIFFERENT TRACK
 5738  025342  001004                          BNE     42$             ;BR IF YES
 5739  025344  123737  001200  005473          CMPB    $REG7,FINSEC    ;SEE IF DIFFERENT SECTOR
 5740  025352  001412                          BEQ     44$             ;BR IF SAME PACK ADDRESS
 5741  025354  013737  005470  001174  42$:    MOV     FINCYL,$REG5    ;SET NEW PACK ADRS FOR PRINTOUT
 5742  025362  113737  005472  001176          MOVB    FINTRK,$REG6
 5743  025370  113737  005473  001200          MOVB    FINSEC,$REG7
 5744  025376  104115                          ERROR   115             ;TYPE NEW PACK ADDRESS
 5745  025400  005437  001202          44$:    NEG     $REG10          ;GET WORD NO.
 5746  025404  016337  177776  001204          MOV     -2(R3),$REG11   ;GET GOOD DATA
 5747  025412  016137  177776  001206          MOV     -2(R1),$REG12   ;GET BAD DATA
 5748  025420  013737  001202  001212          MOV     $REG10,$REG14   ;COMPUTE PHYSICAL ADDRESS
 5749  025426  005037  001210                  CLR     $REG13
 5750  025432  006137  001212                  ROL     $REG14
 5751  025436  006137  001210                  ROL     $REG13
 5752  025442  063737  005504  001212          ADD     PMA,$REG14
 5753  025450  005537  001210                  ADC     $REG13
 5754  025454  063737  005506  001210          ADD     PMA+2,$REG13
 5755  025462  010137  001214                  MOV     R1,$REG15       ;GET VIRT. ADRS FOR PRINTOUT
 5756  025466  162737  000002  001214          SUB     #2,$REG15
 5757  025474  104063                          ERROR   63              ;TYPE GOOD AND BAD DATA, MEM. ADRS.
 5758  025476  032777  000100  153434          BIT     #BIT6,@SWR      ;SEE IF JUST 1 ERROR SHOULD BE REPORTED
 5759  025504  001013                          BNE     54$             ;BR IF JUST 1 ERROR SHOULD BE REPORTED
 5760  025506  005737  005446          50$:    TST     PATRN           ;SEE IF DEFAULT DATA TEST
 5761  025512  001400                          BEQ     40$             ;BR IF YES
 5762  025514  005302                  40$:    DEC     R2              ;DECREMENT WORD COUNTER
 5763  025516  001406                          BEQ     54$             ;BR IF ALL DONE
 5764  025520  005304                          DEC     R4              ;DECREMENT PATTERN WORD COUNT
 5765  025522  001002                          BNE     53$             ;BR IF NOT DONE WITH PATTERN YET
 5766  025524  000137  025144                  JMP     30$             ;JUMP TO REPEAT THE PATTERN
 5767  025530  000137  025150          53$:    JMP     34$
 5768  025534                          54$:
 5769  025534  005726                  56$:    TST     (SP)+           ;POP THE STACK
 5770  025536  104410                          RESREG                  ;RESTORE R0-R5
 5771  025540  000207                          RTS     PC              ;RETURN
 5772
 5773
 5774
 5775                                  ;;*****************************************************************
 5776                                  ;* CTLOUT - THIS SUBROUTINE CHECKS FOR (^C) OR (^Z) TTY INPUT.
 5777                                  ;*IF (^C) WAS TYPED, THE SUBROUTINE RETURNS TO DRVTST. IF (^Z)
 5778                                  ;*WAS TYPED, THE SUBROUTINE RETURNS TO INPUTP. IN EITHER CASE,
 5779                                  ;*A RESET INSTRUCTION IS EXECUTED, AND THE STACK IS RE-INIT-
 5780                                  ;*IALIZED. IF THERE IS NO INPUT, OR INPUT OTHER THAN (^C) OR (^Z),
 5781                                  ;*NO ACTION IS TAKEN.
 5782                                  ;*  CALL -   JSR     PC,CTLOUT
 5783                                  ;*    OR -   CKEXIT
 5784                                  ;;*****************************************************************
 5785  025542  122737  000001  003127  CTLOUT: CMPB    #1,TSTING       ;SEE IF CURRENTLY RUNNING TESTS
 5786  025550  001075                          BNE     10$             ;BR IF NOT RUNNING TESTS
 5787  025552  005737  005442                  TST     INTCHR          ;SEE IF ANY TTY INPUT
 5788  025556  001472                          BEQ     10$             ;BR IF NO INPUT
 5789  025560  105737  003126                  TSTB    MDFLAG          ;SEE IF DEFAULT MODE RUN
 5790  025564  001441                          BEQ     12$             ;BR IF YES
 5791  025566  122737  000003  005442          CMPB    #003,INTCHR     ;SEE IF (^C) TYPED
```

```
 5792  025574  001026                          BNE     4$                  ;BR IF NOT (^C)
 5793  025576  012700  012752                  MOV     #ALLSYS,R0          ;SET RETURN ADDR = ALLSYS
 5794  025602                          2$:
 5795  025602  000005                  6$:     RESET                       ;RESET ALL DEVICES
 5796  025604  005037  005442                  CLR     INTCHR              ;CLEAR TTY CHAR BUFFER WORD
 5797  025610  105037  003127                  CLRB    TSTING              ;CLEAR TTY ESCAPE ALLOW FLAG
 5798  025614  012737  031472  003056          MOV     #ERRHDL,A.ABNL      ;RESTORE ERROR HANDLER ADDRESS
 5799  025622  012706  001100                  MOV     #STACK,SP           ;RESET THE STACK
 5800  025626  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;SET RECAL COMMAND
 5801  025634  004737  030124                  JSR     PC,DRVCAL           ;DO CLEANUP RECALIBRATE
 5802  025640  105037  003143                  CLRB    TSTTYP              ;CLEAR OFFSET TEST IDENTIFIER
 5803  025644  005037  001102                  CLR     $TSTNM              ;CLEAR THE TEST NO.
 5804  025650  000110                          JMP     @R0                 ;EXIT FROM TESTS
 5805  025652  122737  000032  005442  4$:     CMPB    #032,INTCHR         ;SEE IF (^Z) TYPED
 5806  025660  001021                          BNE     7$                  ;BR IF NOT (^Z)
 5807  025662  012700  013416                  MOV     #ASKSYS,R0          ;SET RETURN ADDR = ASKSYS
 5808  025666  000745                          BR      2$                  ;TAKE EXIT
 5809  025670  122737  000003  005442  12$:    CMPB    #003,INTCHR         ;SEE IF (^C) TYPED
 5810  025676  001012                          BNE     7$                  ;BR IF NOT
 5811  025700  104401  011537                  TYPE    ,HLTRQD             ;TYPE "HALT REQUESTED"
 5812  025704  104401  011507                  TYPE    ,CNTRDY             ;TYPE "PRESS CONT WHEN RDY"
 5813  025710  012700  033530                  MOV     #HLTPRG,R0          ;SET HALT ADDRESS
 5814  025714  112737  000061  003155          MOVB    #'1,PASSNO          ;RESTORE PASS NO. TO 1
 5815  025722  000727                          BR      2$                  ;TAKE EXIT
 5816  025724  122737  000007  005442  7$:     CMPB    #007,INTCHR         ;SEE IF (^G) TYPED
 5817  025732  001002                          BNE     8$                  ;BR IF NOT (^G)
 5818  025734  004737  021106                  JSR     PC,GTSWRG           ;OPEN SOFTWARE SWR FOR MODIFICATION
 5819  025740  004737  020744          8$:     JSR     PC,PREPKB           ;ENABLE KBD INPUT AGAIN
 5820  025744  000207                  10$:    RTS     PC                  ;RETURN
 5821
 5822
 5823
 5824                                  ;;*****************************************************************
 5825                                  ;*WRTSEC - WRITE A BLOCK OF 400(OCT) WORDS ONTO THE DISK.
 5826                                  ;*THE PARAMETER BLOCK MUST BE PRE-LOADED WITH THE PACK
 5827                                  ;*ADDRESS. DATA WORD IS PASSED IN R3 ON ENTRY. RWBUF IS THE BUFFER
 5828                                  ;*USED.
 5829                                  ;;*****************************************************************
 5830  025746  104407                  WRTSEC: SAVREG                      ;SAVE R0-R5
 5831                                  ;LOAD THE R/W BUFFER WITH DATA
 5832  025750  012700  053330                  MOV     #RWBUF,R0           ;BUFFER ADDRESS
 5833  025754  012701  000400                  MOV     #400,R1             ;400(OCT) WORDS
 5834  025760  010320                  4$:     MOV     R3,(R0)+            ;LOAD A BUFFER WORD
 5835  025762  005301                          DEC     R1                  ;DECR COUNTER
 5836  025764  001375                          BNE     4$                  ;BR IF NOT DONE YET
 5837                                  ;SET UP PARAMETERS
 5838  025766  012765  053330  000010          MOV     #RWBUF,P.BALO(R5) ;SET BUS ADDRESS
 5839  025774  012765  177400  000012          MOV     #-400,P.WC(R5)  ;SET WORD COUNT
 5840                                  ;WRITE THE DATA
 5841  026002  112765  000123  000001          MOVB    #WRDATA,P.CMND(R5) ;SET WRITE COMMAND
 5842  026010  004737  030124                  JSR     PC,DRVCAL           ;WRITE THE DATA
 5843  026014  104410                          RESREG                      ;RESTORE R0-R5
 5844  026016  000207                          RTS     PC                  ;RETURN
 5845
 5846
 5847
```

CMPBUF - SOFTWARE COMPARE DATA SEQ 0112

```
5848                                    ;;*****************************************************************
5849                                    ;*LDMEM1 - LOAD MEMORY WITH INCREASING NUMBERS, STARTING WITH 1.
5850                                    ;*LDMEM2 - LOAD MEMORY WITH REPEATED WORD, IN R3 ON ENTRY.
5851                                    ;*BOTH SUBROUTINES REQUIRE MEMORY ADDRESS IN PMA, PMA+2, AND WORD COUNT
5852                                    ;*MUST BE IN WDSXFR.
5853                                    ;;*****************************************************************
5854  026020  104407                    LDMEM1: SAVREG                  ;SAVE R0-R5
5855  026022  005004                            CLR     R4               ;R4=0 INDICATES LDMEM1
5856  026024  012703  000001                    MOV     #1,R3            ;INIT DATA WORD TO 1
5857  026030  000403                            BR      LDM0             ;PROCEED
5858  026032  104407                    LDMEM2: SAVREG                  ;SAVE R0-R5
5859  026034  012704  177777                    MOV     #-1,R4           ;R4=177777 INDICATES LDMEM2
5860  026040  013702  005466            LDM0:   MOV     WDSXFR,R2        ;GET NO. OF WORDS
5861  026044  013701  005504                    MOV     PMA,R1           ;GET MEM ADRS
5862  026050  010321                    6$:     MOV     R3,(R1)+         ;LOAD A WORD INTO BUFFER
5863  026052  005704                            TST     R4               ;SEE WHICH DATA DESIRED
5864  026054  001001                            BNE     8$               ;BR IF NOT INCREMENTING DATA
5865  026056  005203                            INC     R3               ;INCREMENT THE DATA
5866  026060                            8$:
5867  026060  005302                    10$:    DEC     R2               ;DECREMENT COUNTER
5868  026062  001372                            BNE     6$               ;BR IF NOT DONE LOADING YET
5869  026064  104410                    12$:    RESREG                   ;RESTORE R0-R5
5870  026066  000207                            RTS     PC               ;RETURN
5871
5872
5873
5874                                    ;;*****************************************************************
5875                                    ;*CKMEM1 - PERFORM SOFTWARE COMPARE OF MEMORY, WITH INCREASING
5876                                    ;*NUMBERS, STARTING WITH 1.
5877                                    ;*CKMEM2 - PERFORM SOFTWARE COMPARE OF MEMORY, WITH REPEATED DATA WORD
5878                                    ;*IN R3 ON ENTRY.
5879                                    ;*BOTH SUBROUTINES REQUIRE MEM ADRS IN PMA, PMA+2, AND WORD COUNT MUST BE
5880                                    ;*IN WDSXFR.
5881                                    ;;*****************************************************************
5882  026070  104407                    CKMEM1: SAVREG                  ;SAVE R0-R5
5883  026072  005004                            CLR     R4               ;R4=0 INDICATES CKMEM1
5884  026074  012703  000001                    MOV     #1,R3            ;INIT DATA TO 1
5885  026100  000403                            BR      CKM0             ;PROCEED
5886  026102  104407                    CKMEM2: SAVREG                  ;SAVE R0-R5
5887  026104  012704  177777                    MOV     #-1,R4           ;R4=177777 INDICATES CKMEM2
5888  026110  013702  005466            CKM0:   MOV     WDSXFR,R2        ;GET NO. OF WORDS
5889  026114  013701  005504                    MOV     PMA,R1           ;GET MEM ADRS
5890  026120  005037  005444                    CLR     SCRACH
5891  026124  112737  000040  051653          MOVB    #40,DH701+38.    ;RESTORE ERROR MSG PARAMS
5892  026132  012737  000007  053222          MOV     #7,DF25+2
5893  026140  004737  031212                  JSR     PC,REPSUP        ;STORE PREV CMND FOR POSS. PRINT
5894  026144  020321                    6$:     CMP     R3,(R1)+         ;COMPARE A DATA WORD
5895  026146  001552                            BEQ     16$              ;BR IF DATA COMPARES OK
5896                                    ;COMPARE ERROR HANDLER
5897  026150  010237  001202                    MOV     R2,$REG10        ;GET WORD NO.
5898  026154  163737  005466  001202          SUB     WDSXFR,$REG10
5899  026162  013737  001202  005474          MOV     $REG10,LASTWC    ;GET 2'S COMP OF WORD NO.
5900  026170  004737  024520                  JSR     PC,FINADR        ;COMPUTE ACTUAL PACK ADRS
5901  026174  104407                          SAVREG                   ;SAVE R0-R5
5902  026176  013700  005470                  MOV     FINCYL,R0        ;GET CYL
5903  026202  113701  005472                  MOVB    FINTRK,R1        ;GET TRACK
```

```
 5904  026206  113702  005473                  MOVB    FINSEC,R2       ;GET SECTOR
 5905  026212  004737  030020                  JSR     PC,BDSRCK       ;SEE IF THIS SECTOR LISTED BAD
 5906  026216  104410                          RESREG                  ;RESTORE R0-R5
 5907  026220  032737  001000  005424          BIT     #BADSEC,RECODE
 5908  026226  001122                          BNE     16$             ;BR IF LISTED- DON'T REPORT ERROR
 5909  026230  105037  051653                  CLRB    DH701+38.       ;ADJUST DATA HEADER FOR MSG
 5910  026234  012737  000005  053222          MOV     #5,DF25+2       ;ADJ. ERROR DATA WORD COUNT
 5911  026242  005737  005444          9$:     TST     SCRACH          ;SEE IF FIRST ERROR IN THIS BLOCK
 5912  026246  001010                          BNE     10$             ;BR IF NOT FIRST ERROR
 5913  026250  104034                  17$:    ERROR   34              ;TYPE HEADING FOR ERROR MSG
 5914  026252  012737  177777  001174  11$:    MOV     #-1,$REG5       ;INIT CYL NO.
 5915  026260  005037  001176                  CLR     $REG6
 5916  026264  005037  001200                  CLR     $REG7
 5917  026270  005237  005444          10$:    INC     SCRACH          ;INCREMENT THE ERROR COUNT
 5918  026274  032777  000001  152636          BIT     #BIT0,@SWR      ;SEE IF ALL ERRORS SHOULD BE REPORTED
 5919  026302  001004                          BNE     12$             ;BR TO REPORT ALL ERRORS
 5920  026304  022737  000012  005444          CMP     #10.,SCRACH     ;SEE IF 10(DEC) ERRORS YET
 5921  026312  002477                          BLT     21$             ;BR IF ERROR LIMIT EXCEEDED
 5922  026314  023737  001174  005470  12$:    CMP     $REG5,FINCYL    ;SEE IF DIFFERENT CYL
 5923  026322  001010                          BNE     14$             ;BR IF YES
 5924  026324  123737  001176  005472          CMPB    $REG6,FINTRK    ;SEE IF DIFFERENT TRACK
 5925  026332  001004                          BNE     14$             ;BR IF YES
 5926  026334  123737  001200  005473          CMPB    $REG7,FINSEC    ;SEE IF DIFFERENT SECTOR
 5927  026342  001412                          BEQ     15$             ;BR IF SAME PACK ADDRESS
 5928  026344  013737  005470  001174  14$:    MOV     FINCYL,$REG5    ;SET NEW PACK ADR FOR PRINTOUT
 5929  026352  113737  005472  001176          MOVB    FINTRK,$REG6
 5930  026360  113737  005473  001200          MOVB    FINSEC,$REG7
 5931  026366  104115                          ERROR   115             ;TYPE NEW PACK ADRS
 5932  026370  005437  001202          15$:    NEG     $REG10          ;GET WORD NO.
 5933  026374  010337  001204                  MOV     R3,$REG11       ;GOOD DATA
 5934  026400  016137  177776  001206          MOV     -2(R1),$REG12   ;BAD DATA
 5935  026406  013737  001202  001212          MOV     $REG10,$REG14   ;COMPUTE PHYSICAL ADDRESS
 5936  026414  005037  001210                  CLR     $REG13
 5937  026420  006137  001212                  ROL     $REG14
 5938  026424  006137  001210                  ROL     $REG13
 5939  026430  063737  005504  001212          ADD     PMA,$REG14
 5940  026436  005537  001210                  ADC     $REG13
 5941  026442  063737  005506  001210          ADD     PMA+2,$REG13
 5942  026450  010137  001214                  MOV     R1,$REG15       ;GET VIRT ADRS FOR PRINTOUT
 5943  026454  162737  000002  001214          SUB     #2,$REG15
 5944  026462  104063                          ERROR   63              ;TYPE GOOD AND BAD DATA
 5945  026464  032777  000100  152446          BIT     #BIT6,@SWR      ;SEE IF JUST 1 ERROR SHOULD BE REPORTED
 5946  026472  001007                          BNE     21$             ;BR IF JUST 1
 5947                          ;PROCEED WITH COMPARISONS
 5948  026474  005704                  16$:    TST     R4              ;SEE WHICH DATA DESIRED
 5949  026476  001001                          BNE     18$             ;BR IF NOT INCREASING DATA
 5950  026500  005203                          INC     R3              ;INCREMENT THE DATA WORD
 5951  026502                          18$:
 5952  026502  005302                  20$:    DEC     R2              ;DECR COUNTER
 5953  026504  001402                          BEQ     21$             ;BR IF DONE COMPARING
 5954  026506  000137  026144                  JMP     6$              ;CONTINUE COMPARING WORDS
 5955  026512                          21$:
 5956  026512  104410                  22$:    RESREG                  ;RESTORE R0-R5
 5957  026514  000207                          RTS     PC              ;RETURN
 5958
 5959
```

```
 5960
 5961
 5962
 5963                                   ;;*****************************************************************
 5964                                   ;*REDBSF - READ FACTORY BSF INTO BSFACT, AND READ SOFTWARE BSF
 5965                                   ;*INTO BSSOFT.
 5966                                   ;;*****************************************************************
 5967  026516                           REDBSF:
 5968  026516  004737  021140                   JSR     PC,INITSS       ;INIT THE S.S.
 5969  026522  142765  000020  000007           BICB    #B.CFMT,P.CS1H(R5)      ;SET 22 SECTOR FORMAT
 5970  026530  112765  000121  000001           MOVB    #RDDATA,P.CMND(R5) ;SET READ DATA COMMAND
 5971  026536  013765  024362  000002           MOV     LSTCYL,P.CYLN(R5)       ;SET CYL = 632/1456
 5972  026544  112765  000002  000005           MOVB    #2,P.TRCK(R5)   ;SET TRACK = 2
 5973  026552  012703  000012                   MOV     #10.,R3         ;SET FACTORY BSF SECTOR LIMIT
 5974  026556  105737  003134                   TSTB    FORMAT
 5975  026562  001403                           BEQ     6$              ;BR IF 22 SECTORS
 5976  026564  105265  000004                   INCB    P.SECT(R5)      ;SET STARTING FACTORY BSF SECTOR NO.
 5977  026570  005203                           INC     R3
 5978  026572  012765  177400  000012  6$:      MOV     #-400,P.WC(R5)  ;SET WORD COUNT FOR 1 SECTOR
 5979  026600  012765  004162  000010           MOV     #BSFACT,P.BALO(R5) ;SET BA FOR FACTORY BSF
 5980  026606  012737  026762  003056  8$:      MOV     #BDSCHD,A.ABNL ;SET ERROR HANDLER FOR BSF READ
 5981  026614  105037  003145                   CLRB    WCEFLG          ;INIT THE ERROR FLAG
 5982  026620  004737  030124                   JSR     PC,DRVCAL       ;READ THE FACTORY BSF
 5983  026624  105737  003145                   TSTB    WCEFLG          ;SEE IF ANY ERRORS
 5984  026630  001413                           BEQ     12$             ;BR IF NOT
 5985  026632  062765  000002  000004           ADD     #2,P.SECT(R5)   ;CHECK NEXT SECTOR
 5986  026640  126503  000004                   CMPB    P.SECT(R5),R3   ;SEE IF LIMIT EXCEEDED YET
 5987  026644  001360                           BNE     8$              ;BR IF NOT YET
 5988  026646  004737  031212          10$:     JSR     PC,REPSUP       ;GATHER STATUS FOR PRINTOUT
 5989  026652  104120                           ERROR   120             ;ABORTING- BAD BSF READ
 5990  026654  000137  033530                   JMP     HLTPRG          ;HALT- CAN'T PROCEED
 5991  026660  012765  003162  000010  12$:     MOV     #BSSOFT,P.BALO(R5) ;SET BA FOR SOFTWARE RECORD
 5992  026666  110365  000004                   MOVB    R3,P.SECT(R5)   ;SET STARTING SECTOR NO.
 5993  026672  012703  000026                   MOV     #22.,R3         ;SET 22 SECTOR LIMIT
 5994  026676  105737  003134                   TSTB    FORMAT          ;SEE IF 22 SECTOR FORMAT
 5995  026702  001402                           BEQ     14$             ;BR IF YES
 5996  026704  012703  000023                   MOV     #19.,R3         ;SET 20 SECTOR LIMIT
 5997  026710  012737  026762  003056  14$:     MOV     #BDSCHD,A.ABNL ;SET ERROR HANDLER FOR BSF READ
 5998  026716  105037  003145                   CLRB    WCEFLG          ;INIT THE ERROR FLAG
 5999  026722  004737  030124                   JSR     PC,DRVCAL       ;READ THE SOFTWARE BSF
 6000  026726  105737  003145                   TSTB    WCEFLG          ;SEE IF ANY ERRORS
 6001  026732  001407                           BEQ     16$             ;BR IF NOT
 6002  026734  062765  000002  000004           ADD     #2,P.SECT(R5)   ;CHECK NEXT SECTOR
 6003  026742  126503  000004                   CMPB    P.SECT(R5),R3   ;SEE IF LIMIT EXCEEDED YET
 6004  026746  003760                           BLE     14$             ;BR IF NOT YET
 6005  026750  000736                           BR      10$             ;REPORT ERROR AND ABORT
 6006  026752  012737  031472  003056  16$:     MOV     #ERRHDL,A.ABNL  ;RESTORE ERROR HANDLER ADRS
 6007  026760  000207                           RTS     PC              ;RETURN
 6008
 6009                                   ;*THIS IS THE ABNORMAL DRIVER RETURN WHICH IS USED TO HANDLE
 6010                                   ;*POSSIBLE ERRORS IN READING BAD SECTORS.
 6011  026762  032765  130600  000034  BDSCHD: BIT     #BSE!HVRC!DTE!OPI!DCK,P.ER(R5) ;SEE IF ANY READ ERRORS
 6012  026770  001002                           BNE     4$              ;BR IF A READ ERROR OCCURRED
 6013  026772  000137  031472                   JMP     ERRHDL          ;GO HANDLE OTHER ERROR
 6014  026776  105237  003145          4$:      INCB    WCEFLG          ;SET ERROR FLAG
 6015  027002  000137  033742                   JMP     RETNML          ;TAKE NORMAL DRIVER RETURN
```

```
                                        ;;*****************************************************************
                                        ;*TYPBSF - TYPE FACTORY AND SOFTWARE BAD SECTOR FILES (BSF'S)
                                        ;*WITH THE CYLINDER, TRACK, AND SECTOR NO. OF EACH BAD SECTOR
                                        ;*LISTED SEPARATELY, IN A TABLE OF OCTAL VALUES, FOR EACH OF THE
                                        ;*2 BSF'S.
                                        ;;*****************************************************************
027006  104407                  TYPBSF: SAVREG                  ;SAVE R0-R5
027010  005004                          CLR     R4              ;INIT BSF FILE INDICATOR
027012  012700  004172                  MOV     #BSFACT+10,R0   ;ADRS OF DATA IN FACTORY BSF
027016  104401  011732                  TYPE    ,FACTBS         ;TYPE "FACTORY"
027022  104401  011757                  TYPE    ,BDSECT         ;TYPE "BAD SECTORS"
027026  104401  051420          3$:     TYPE    ,DH6041         ;TYPE   "CYLNDR  TRACK   SECTOR"
027032  104401  001325                  TYPE    ,$CRLF
027036  005001                          CLR     R1              ;INIT LIMIT COUNTER
027040  005710                  4$:     TST     (R0)            ;SEE IF A SECTOR LISTED HERE
027042  100007                          BPL     8$              ;BR IF YES
027044  005701                          TST     R1              ;SEE IF FILE IS EMPTY
027046  001032                          BNE     14$             ;BR IF NOT EMPTY
027050  104401  011777                  TYPE    ,NOFALS         ;TYPE "NONE"
027054  104401  001325                  TYPE    ,$CRLF          ;TYPE <CR>,<LF>
027060  000425                          BR      14$
027062  012046                  8$:     MOV     (R0)+,-(SP)     ;GET A CYL NO.
027064  104402                          TYPOC                   ;TYPE IT
027066  104401  012117                  TYPE    ,SPACE2         ;TYPE SEPARATORS
027072  011003                          MOV     (R0),R3         ;GET TRACK AND SECTOR
027074  105003                          CLRB    R3
027076  000303                          SWAB    R3              ;GET THE TRACK NO.
027100  010346                          MOV     R3,-(SP)
027102  104402                          TYPOC                   ;TYPE IT
027104  104401  012117                  TYPE    ,SPACE2         ;TYPE SEPARATORS
027110  111003                          MOVB    (R0),R3         ;GET SECTOR NO.
027112  010346                          MOV     R3,-(SP)
027114  104402                          TYPOC                   ;TYPE IT
027116  104401  001325                  TYPE    ,$CRLF
027122  005720                          TST     (R0)+           ;INCR THE POINTER
027124  005201                          INC     R1              ;INCR THE COUNTER
027126  020127  000176                  CMP     R1,#126.        ;SEE IF LIMIT REACHED IN THIS FILE
027132  002742                          BLT     4$              ;BR IF NOT YET
027134  005704                  14$:    TST     R4              ;SEE IF BOTH FILES TYPED YET
027136  001006                          BNE     18$             ;BR IF YES
027140  005204                          INC     R4              ;SET INDICATOR FOR SOFT. FILE
027142  012700  003172                  MOV     #BSSOFT+10,R0   ;ADRS OF DATA IN SOFTWARE BSF
027146  104401  011745                  TYPE    ,SOFTBS         ;TYPE "SOFTWARE BAD SECTORS :"
027152  000725                          BR      3$              ;GO TYPE SOFT. BSF
027154  104410                  18$:    RESREG                  ;RESTORE R0-R5
027156  000207                          RTS     PC              ;RETURN


                                        ;;*****************************************************************
                                        ;*FINMEM - COMPUTE NO. OF WORDS XFERRED ON PARTIAL XFER
                                        ;*(TERMINATED BY BSE ERROR), AND STORE IT IN WDSXFR.
                                        ;;*****************************************************************
```

```
 6072  027160  104407                 FINMEM: SAVREG                          ;SAVE R0-R5
 6073  027162  016500  000002                 MOV     P.CYLN(R5),R0          ;GET ORIG. CYL NO.
 6074  027166  116501  000005                 MOVB    P.TRCK(R5),R1          ;GET TRACK NO.
 6075  027172  116502  000004                 MOVB    P.SECT(R5),R2          ;SECTOR NO.
 6076  027176  005003                         CLR     R3                     ;INIT SECTOR COUNT TO 0
 6077  027200  026500  000030         6$:     CMP     P.DCYL(R5),R0          ;COMPARE ORIG. AND CURRENT CYLS
 6078  027204  001006                         BNE     8$                     ;BR IF NOT EQUAL
 6079  027206  126501  000027                 CMPB    P.DTS+1(R5),R1         ;COMPARE TRACKS
 6080  027212  001003                         BNE     8$                     ;BR IF NOT EQUAL
 6081  027214  126502  000026                 CMPB    P.DTS(R5),R2           ;COMPARE SECTORS
 6082  027220  001404                         BEQ     12$                    ;BR IF ADRS ARE EQUAL
 6083  027222  005203                 8$:     INC     R3                     ;INCR SECTOR COUNT
 6084  027224  004737  027244                 JSR     PC,INCRSC              ;INCR PACK ADRS BY 1 SECTOR
 6085  027230  000763                         BR      6$                     ;GO COMPARE ADDRESSES
 6086  027232  000303                 12$:    SWAB    R3                     ;COMPUTE NO. OF WORDS XFERRED
 6087  027234  010337  005466                 MOV     R3,WDSXFR              ;STORE IT
 6088  027240  104410                         RESREG                         ;RESTORE R0-R5
 6089  027242  000207                         RTS     PC                     ;RETURN
 6090
 6091
 6092
 6093                                 ;;*****************************************************************
 6094                                 ;*INCRSC - INCREMENT PACK ADDRESS TO NEXT SECTOR
 6095                                 ;*THE CYLINDER IS PASSED AND RETURNED IN R0, TRACK IN R1, AND SECTOR
 6096                                 ;*IN R2.
 6097                                 ;;*****************************************************************
 6098  027244  012704  000025        INCRSC: MOV     #21.,R4                ;SET SECTOR LIMIT
 6099  027250  105737  003134                 TSTB    FORMAT                 ;SEE IF 22 SECTOR FORMAT
 6100  027254  001402                         BEQ     4$                     ;BR IF YES
 6101  027256  012704  000023                 MOV     #19.,R4                ;SET SECTOR LIMIT
 6102  027262  005202                 4$:     INC     R2                     ;INCR. SECTOR NO.
 6103  027264  020204                         CMP     R2,R4                  ;SEE IF SECTOR LIMIT EXCEEDED
 6104  027266  003407                         BLE     6$                     ;BR IF NOT
 6105  027270  005002                         CLR     R2                     ;SET SECTOR = 0
 6106  027272  005201                         INC     R1                     ;INCR. TRACK NO.
 6107  027274  020127  000002                 CMP     R1,#2                  ;SEE IF TRACK LIMIT EXCEEDED
 6108  027300  003402                         BLE     6$                     ;BR IF NOT
 6109  027302  005001                         CLR     R1                     ;SET TRACK = 0
 6110  027304  005200                         INC     R0                     ;INCR. CYLINDER NO.
 6111  027306  000207                 6$:     RTS     PC                     ;RETURN
 6112
 6113
 6114
 6115                                 ;;*****************************************************************
 6116                                 ;*MIDXFR - MID-TRANSFER PARAMETER UPDATE SUBROUTINE
 6117                                 ;*THIS SUBROUTINE UPDATES PMA, PMA+2, P.BALO(R5),
 6118                                 ;*P.BAHI(R5), AND P.WC(R5), IN PREPARATION TO RESUME A TRANSFER
 6119                                 ;*TERMINATED BY A BAD SECTOR ERROR (BSE), A DATA CHECK ERROR (DCK),
 6120                                 ;*OR A HEADER VRC ERROR (HVRC).
 6121                                 ;;*****************************************************************
 6122  027310  104407                 MIDXFR: SAVREG                         ;SAVE R0-R5
 6123  027312  004737  027160                 JSR     PC,FINMEM              ;COMPUTE NO. OF WORDS XFERRED
 6124  027316  016500  000030                 MOV     P.DCYL(R5),R0          ;GET CYL NO.
 6125  027322  116501  000027                 MOVB    P.DTS+1(R5),R1         ;GET TRACK NO.
 6126  027326  116502  000026                 MOVB    P.DTS(R5),R2           ;GET SECTOR NO.
 6127  027332  013703  005466                 MOV     WDSXFR,R3              ;GET NO. OF WORDS TRANSFERRED
```

```
 6128  027336  032757  000020  005424          BIT     #DCKERR,RECODE  ;SEE IF DATA CHECK ERROR OCCURRED
 6129  027344  001004                          BNE     9$              ;BR IF YES
 6130  027346  062703  000400                  ADD     #400,R3         ;SKIP BAD SECTOR
 6131  027352  004737  027244                  JSR     PC,INCRSC       ;INCREMENT PACK ADRS TO NEXT SECTOR
 6132  027356  010065  000002          9$:     MOV     R0,P.CYLN(R5)   ;UPDATE CYLINDER
 6133  027362  110165  000005                  MOVB    R1,P.TRCK(R5)   ;UPDATE TRACK
 6134  027366  110265  000004                  MOVB    R2,P.SECT(R5)   ;UPDATE SECTOR
 6135  027372  060365  000012                  ADD     R3,P.WC(R5)     ;UPDATE P.WC(R5)
 6136  027376  032765  100000  000014          BIT     #DTBAII,P.PRST(R5) ;SEE IF BUS ADRS INCR INHIB SET
 6137  027404  001022                          BNE     16$             ;BR IF YES
 6138  027406  005004                          CLR     R4              ;GET BYTES XFERRED
 6139  027410  006103                          ROL     R3
 6140  027412  006104                          ROL     R4
 6141  027414  060337  005504                  ADD     R3,PMA          ;UPDATE PMA,PMA+2
 6142  027420  005537  005506                  ADC     PMA+2
 6143  027424  060437  005506                  ADD     R4,PMA+2
 6144  027430  013765  005504  000010          MOV     PMA,P.BALO(R5)  ;UPDATE P.BALO(R5)
 6145  027436  013700  005506                  MOV     PMA+2,R0
 6146  027442  042700  177774                  BIC     #177774,R0
 6147  027446  150065  000007                  BISB    R0,P.BAHI(R5)   ;UPDATE P.BAHI(R5)
 6148  027452  104410                  16$:    RESREG                  ;RESTORE R0-R5
 6149  027454  000207                          RTS     PC              ;RETURN
 6150
 6151
 6152
 6153                                  ;;*****************************************************************
 6154                                  ;*SVPRMS - SAVE INITIAL PARAMETERS FOR TRANSFER
 6155                                  ;*GTPRMS - RESTORE SAVED TRANSFER PARAMETERS
 6156                                  ;;*****************************************************************
 6157  027456  104407                  SVPRMS: SAVREG                  ;SAVE R0-R5
 6158  027460  012700  002642                  MOV     #PARM0+2,R0     ;ADRS OF PARAMS
 6159  027464  012701  005454                  MOV     #SAVPRS,R1      ;ADRS OF SAVE AREA
 6160  027470  016537  000012  005466          MOV     P.WC(R5),WDSXFR  ;GET WORD COUNT
 6161  027476  005437  005466                  NEG     WDSXFR          ;MAKE IT POSITIVE
 6162  027502  012737  053330  005504          MOV     #RWBUF,PMA      ;INIT PMA TO RWBUF
 6163  027510  005037  005506                  CLR     PMA+2           ;INIT PMA+2 TO 0
 6164  027514  000405                          BR      GTO             ;CONTINUE
 6165  027516  104407                  GTPRMS: SAVREG                  ;SAVE R0-R5
 6166  027520  012700  005454                  MOV     #SAVPRS,R0      ;ADRS OF SAVE AREA
 6167  027524  012701  002642                  MOV     #PARM0+2,R1     ;ADRS OF PARAMS
 6168  027530  012702  000005          GTO:    MOV     #5,R2           ;SET FOR 5 WORDS
 6169  027534  012021                  6$:     MOV     (R0)+,(R1)+     ;MOVE A WORD
 6170  027536  005302                          DEC     R2              ;SEE IF DONE YET
 6171  027540  001375                          BNE     6$              ;BR IF NOT YET
 6172  027542  104410                          RESREG                  ;RESTORE R0-R5
 6173  027544  000207                          RTS     PC              ;RETURN
 6174
 6175
 6176
 6177                                  ;;*****************************************************************
 6178                                  ;*TRNSFR - PERFORM A DATA TRANSFER, AND HANDLE BAD SECTORS
 6179                                  ;*OR DATA CHECK ERRORS, IF ENCOUNTERED.
 6180                                  ;*THE COMMAND MUST BE SET IN THE PARAM BLK. THIS SUB-
 6181                                  ;*ROUTINE IS SUBSTITUTED FOR DRVCAL WHEN BAD SECTORS MUST BE HANDLED,
 6182                                  ;*AND IT IS ALSO USED TO HANDLE THE DATA CHECK ERRORS CAUSED
 6183                                  ;*BY READING DATA WITH OFFSETS, IN THE OVERWRITE AND DATA COMPATIBILITY
```

```
 6184                                   ;*TESTS.
 6185                                   ;;*****************************************************************
 6186  027546  010446                   TRNSFR: MOV     R4,-(SP)        ;SAVE R4 ON STACK
 6187  027550  005004                           CLR     R4              ;CLEAR ERROR INDICATOR
 6188  027552  105037  003144                   CLRB    DCEFLG          ;CLEAR DCK ERROR FLAG
 6189  027556  005037  005424           4$:     CLR     RECODE          ;CLEAR ERROR FLAGS
 6190  027562  012737  020526  003056           MOV     #DCKHDL,A.ABNL  ;SET DCK ERROR HANDLER ADRS
 6191  027570  105037  003135                   CLRB    ERRCNT          ;CLEAR ERROR RETRY COUNT
 6192  027574  004737  030124                   JSR     PC,DRVCAL       ;PERFORM THE S.S. FUNCTION
 6193  027600  032737  000026  005424           BIT     #BSERR!DCKERR!HVRCER,RECODE  ;SEE IF BSE OR DCK OR HVRC ERROR OCCURRED
 6194  027606  001410                           BEQ     6$              ;BR IF NOT
 6195  027610  005204                           INC     R4              ;INCR ERROR INDICATOR
 6196  027612  004737  023176                   JSR     PC,CKSCOR       ;DECR. DRIVE SCORE, IF NECESSARY
 6197  027616  004737  027310                   JSR     PC,MIDXFR       ;UPDATE PARAMS TO RESUME XFER
 6198  027622  005765  000012                   TST     P.WC(R5)        ;SEE IF ENTIRE XFER IS COMPLETED
 6199  027626  001353                           BNE     4$              ;BR IF NOT
 6200  027630  005704                   6$:     TST     R4              ;SEE IF ANY ERRORS OCCURRED
 6201  027632  001404                           BEQ     8$              ;BR IF NOT
 6202  027634  004737  027516                   JSR     PC,GTPRMS       ;RESTORE ORIGINAL PARAMS OF XFER
 6203  027640  004737  027456                   JSR     PC,SVPRMS       ;RESTORE WDSXFR,PMA,PMA+2
 6204  027644  012604                   8$:     MOV     (SP)+,R4        ;RESTORE R4
 6205  027646  012737  031472  003056           MOV     #ERRHDL,A.ABNL  ;RESTORE NORMAL ERROR HANDLER ADRS
 6206  027654  000207                           RTS     PC              ;RETURN
 6207
 6208
 6209
 6210                                   ;;*****************************************************************
 6211                                   .SBTTL  SEARCH BAD SECTOR TABLES ROUTINE
 6212                                   ;*THIS ROUTINE RETURNS A LIST OF BAD SECTORS FOUND IN
 6213                                   ;*THE SPECIFIC TABLE (BAD SECTOR SOFTWARE OR BAD SECTOR FACTORY)
 6214                                   ;*FOR THE SPECIFIC CYLINDER AND TRACK DESIRED. R0 AND R1 MUST
 6215                                   ;*CONTAIN THE CYLINDER AND TRACK, RESPECTIVELY.
 6216                                   ;*
 6217                                   ;*THE LIST OF BAD SECTORS IS FOUND ON THE STACK WHEN THE ROUTINE
 6218                                   ;*RETURNS TO THE CALLER. THE FIRST ENTRY ON THE STACK WILL BE THE
 6219                                   ;*NUMBER OF BAD SECTORS FOUND FOR THIS CYLINDER AND TRACK.
 6220                                   ;;*****************************************************************
 6221  027656  010237  001266           SRHTBS: MOV     R2,$TMP2
 6222  027662  010337  001270                   MOV     R3,$TMP3
 6223  027666  012637  001272                   MOV     (SP)+,$TMP4     ;STORE RETURN CONTENTS OF R4
 6224  027672  011402                           MOV     (R4),R2 ;GET ADDRESS OF BAD SECTOR TABLE TO SEARCH
 6225  027674  012437  001264                   MOV     (R4)+,$TMP1     ;SETUP FOR LENGTH OF BAD SECTOR TABLE
 6226  027700  062737  001000  001264           ADD     #1000,$TMP1
 6227  027706  005003                           CLR     R3              ;CLEAR R3 FOR COUNT
 6228  027710  062702  000010                   ADD     #10,R2          ;SET R2 FOR POINTER TO CYLINDER ENTRY
 6229  027714  005712                   1$:     TST     (R2)            ;TEST IF ALL ONES
 6230  027716  100430                           BMI     5$              ;YES-DONE
 6231  027720  020012                           CMP     R0,(R2)         ;TEST IF BAD SECTOR IN PRESENT CYL
 6232  027722  001406                           BEQ     3$              ;YES-GO CHECK TRACK
 6233  027724  062702  000004                   ADD     #4,R2           ;ELSE BUMP POINTER
 6234  027730  020237  001264                   CMP     R2,$TMP1        ;TEST IF OUT OF TABLE
 6235  027734  002021                           BGE     5$              ;YES-EXIT
 6236  027736  000766                           BR      1$              ;LOOP
 6237  027740  005722                   3$:     TST     (R2)+           ;TRACK CHECK - PUT POINTER AT TRK/SEC BYTE
 6238  027742  011237  001302                   MOV     (R2),$TMP10     ;GET TRK/SEC WORD
 6239  027746  042737  174377  001302           BIC     #174377,$TMP10  ;CLEAR ALL BUT TRACK
```

```
 6240  027754  123701  001303                  CMPB    $TMP10+1,R1     ;CHECK IF BAD SECTOR IN THIS TRACK
 6241  027760  001402                          BEQ     4$              ;YES - GO PUT SECTOR NUMBER ON STACK
 6242  027762  005722                          TST     (R2)+           ;ELSE BUMP POINTER TO NEXT CYL WORD
 6243  027764  000753                          BR      1$              ;GO TEST NEXT CYL
 6244  027766  005203                  4$:     INC     R3              ;BUMP BAD SECTOR COUNT
 6245  027770  012246                          MOV     (R2)+,-(SP)     ;PUT TRK/SEC WORD ON STACK
 6246  027772  042716  177700                  BIC     #177700,(SP)    ;CLEAR ALL BUT SECTOR NUMBER
 6247  027776  000746                          BR      1$              ;GO CHECK REST OF FILE
 6248  030000  010346                  5$:     MOV     R3,-(SP)        ;EXIT - PUT NUMBER OF BAD
 6249  030002  013702  001266                  MOV     $TMP2,R2        ;SECTORS ON STACK-RESTORE
 6250  030006  013703  001270                  MOV     $TMP3,R3        ;REGISTERS
 6251  030012  013746  001272                  MOV     $TMP4,-(SP)     ;PUT RETURN ON STACK
 6252  030016  000204                          RTS     R4              ;RETURN
 6253                                  ;;*****************************************************************
 6254                                  .SBTTL  BAD SECTOR CHECK ROUTINE
 6255                                  ;*THIS ROUTINE WILL SEARCH BOTH TABLES TO DETERMINE IF A
 6256                                  ;*SPECIFIC SECTOR IS LISTED AS BAD. IF THE SECTOR IS LISTED IN THE
 6257                                  ;*TABLES THE ROUTINE SETS THE "BADSEC" FLAG AND RETURNS. IF THE SECTOR
 6258                                  ;*IS NOT LISTED THE FLAG IS RESET.
 6259                                  ;;*****************************************************************
 6260  030020  104407                  BDSRCK: SAVREG
 6261  030022  012737  004162  030034          MOV     #BSFACT,1$      ;SET TABLE TO SEARCH
 6262  030030  004437  027656          2$:     JSR     R4,SRHTBS       ;GO SEARCH IT
 6263  030034  000000                  1$:     .WORD                   ;TABLE ADDRESS GOES HERE
 6264  030036  012603                          MOV     (SP)+,R3        ;GET NUMBER OF BAD SECTORS, IF ANY
 6265  030040  001015                          BNE     6$              ;IF ANY, GO TEST WHICH ONES
 6266  030042  023727  030034  003162  7$:     CMP     1$,#BSSOFT      ;ELSE TEST IF OTHER TABLE ALREADY SEARCHED
 6267  030050  001404                          BEQ     3$              ;IF YES-EXIT
 6268  030052  012737  003162  030034          MOV     #BSSOFT,1$      ;SET OTHER TABLE FOR SEARCH
 6269  030060  000763                          BR      2$              ;GO SEARCH IT
 6270  030062  042737  001000  005424  3$:     BIC     #BADSEC,RECODE  ;CLEAR BAD SECTOR BIT
 6271  030070  104410                  4$:     RESREG
 6272  030072  000207                          RTS     PC              ;RETURN
 6273  030074  022602                  6$:     CMP     (SP)+,R2        ;THIS SECTOR IN TABLE?
 6274  030076  001403                          BEQ     8$              ;YES-GO SET BIT & EXIT
 6275  030100  005303                          DEC     R3              ;DECREMENT BAD SECTOR COUNT
 6276  030102  001374                          BNE     6$              ;IF NOT ZERO-CHECK NEXT ENTRY
 6277  030104  000756                          BR      7$              ;ELSE GO SEARCH OTHER TABLE
 6278  030106  052737  001000  005424  8$:     BIS     #BADSEC,RECODE  ;SET BAD SECTOR BIT
 6279  030114  005303                  9$:     DEC     R3              ;CLEAR STACK OF OTHER BAD SECTOR
 6280  030116  001764                          BEQ     4$              ;NUMBER
 6281  030120  005726                          TST     (SP)+
 6282  030122  000774                          BR      9$              ;EXIT
 6283
 6284
 6285
 6286
 6287                                  ;;*****************************************************************
 6288                                  .SBTTL  CALL DRIVER ROUTINE
 6289                                  ;*ENTRY JSR PC,DRVCAL
 6290                                  ;*              WITH R5 POINTING TO PARAMETER BLOCK
 6291                                  ;*RETURN RTS PC
 6292                                  ;*
 6293                                  ;*THIS ROUTINE IS USED TO INITIATE A SUBSYSTEM OPERATION BY
 6294                                  ;*CALLING THE DRIVER. THE PARAMETER BLOCK MUST BE SET UP
 6295                                  ;*BY THE CALLER AND R5 MUST POINT TO THE PARAMETER
```

```
                                       ;*BLOCK WHEN THE ROUTINE IS CALLED.
                                       ;*
                                       ;*THIS ROUTINE WAITS FOR THE OPERATION TO BE COMPLETED.
                                       ;*WHILE WAITING THE WATCHDOG TIMER IS CALLED TO PREVENT
                                       ;*SILENT DEATH IN CASE THE SUBSYSTEM DOES NOT PROVIDE AN
                                       ;*INTERRUPT. THE TERMINATION HANDLER ROUTINES WILL SET THE DONE
                                       ;*FLAG WHICH KEYS THE ROUTINE TO RETURN TO THE CALLER.
                                       ;;*****************************************************************
030124                                 DRVCAL:
030124  004737  024376                         JSR     PC,STALL        ;PERFORM A STALL IF REQUIRED
030130  004737  025542                         JSR     PC,CTLOUT       ;CHECK FOR (^C) OR (^Z) KBD INPUT
030134  105037  003132                         CLRB    DONE            ;CLEAR DONE FLAG
030140  105037  003141                         CLRB    DRNAFG          ;CLEAR DRIVE SIEZED BY OTHER PORT FLAG
030144  004737  030176                         JSR     PC,STRCMD       ;STORE PREV AND CURRENT COMMANDS
030150  010537  030160                         MOV     R5,4$           ;GET PARAM BLOCK ADDRESS
030154  004737  040210                         JSR     PC,C.INIT       ;CALL DRIVER
030160  000000                         4$:     .WORD                   ;P.B. ADDRESS GOES HERE
030162  004737  034560                 6$:     JSR     PC,W.WTCH       ;CALL WATCH DOG
030166  105737  003132                         TSTB    DONE            ;DONE SET?
030172  001773                               BEQ     6$              ;NO-LOOP
030174  000207                               RTS     PC              ;YES-RETURN


                                       ;;*****************************************************************
                                       ;*STRCMD - STORE PREVIOUS AND CURRENT SUBSYSTEM COMMANDS
                                       ;;*****************************************************************
030176  104407                         STRCMD: SAVREG                  ;SAVE R0-R5
030200  012701  005162                         MOV     #PRVCMD,R1      ;ADDR OF PREV COMMAND STORAGE
030204  012700  005176                         MOV     #COMSTR,R0      ;ADDR OF CURRENT COMMAND STORAGE
                                       ;STORE PREVIOUS COMMAND
030210  012021                               MOV     (R0)+,(R1)+
030212  012021                               MOV     (R0)+,(R1)+
030214  012021                               MOV     (R0)+,(R1)+
030216  012021                               MOV     (R0)+,(R1)+
030220  012021                               MOV     (R0)+,(R1)+
030222  012021                               MOV     (R0)+,(R1)+
                                       ;STORE CURRENT COMMAND
030224  012701  005176                 4$:     MOV     #COMSTR,R1
030230  012521                               MOV     (R5)+,(R1)+     ;R5 POINTS TO DRIVER PARAM BLK
030232  012521                               MOV     (R5)+,(R1)+
030234  012521                               MOV     (R5)+,(R1)+
030236  012521                               MOV     (R5)+,(R1)+
030240  012521                               MOV     (R5)+,(R1)+
030242  012521                               MOV     (R5)+,(R1)+
030244  104410                               RESREG                  ;RESTORE R0-R5
030246  000207                               RTS     PC              ;RETURN


                                       ;;*****************************************************************
                                       .SBTTL  DRIVE ERROR FREE RETURN ROUTINE
                                       ;*THIS ROUTINE IS CALLED BY THE DRIVER WHEN NO ERROR
                                       ;*HAS BEEN DETECTED IN THE OPERATION. THE ROUTINE SETS THE
                                       ;*DONE FLAG THAT IS TESTED IN THE DRIVER CALLING
                                       ;*ROUTINE.
```

```
 6352                                   ;;*****************************************************************
 6353  030250  *52737  000377  003132  ERRFRE: BISB    #377,DONE        ;SET THE DONE FLAG
 6354  030256  032737  100000  005424          BIT     #ANYDER,RECODE   ;TEST F. ANY DATA ERROR
 6355  030264  001403                          BEQ     2$               ;IF NO - DO ERROR RECOVERY PRINT TEST
 6356  030266  105037  003135                  CLRB    ERRCNT           ;CLEAR ERROR COUNT
 6357  030272  000413                          BR      1$               ;EXIT
 6358  030274  105737  003135          2$:     TSTB    ERRCNT           ;CHECK IF ANY ERRORS HAVE OCCURRED
 6359  030300  001410                          BEQ     1$               ;NO - SKIP TO EXIT
 6360  030302  005037  001174                  CLR     $REG5
 6361  030306  113737  003135  001174          MOVB    ERRCNT,$REG5     ;GET RETRY COUNT
 6362  030314  104101                          ERROR   101              ;PRINT RETRY SUCCESSFUL MESSAGE
 6363  030316  105037  003135                  CLRB    ERRCNT           ;CLEAR ERROR COUNT
 6364  030322  005037  005424          1$:     CLR     RECODE           ;CLEAR RECOVERY FLAGS
 6365  030326  000207                          RTS     PC               ;RETURN
 6366                                   ;;*****************************************************************
 6367                                   .SBTTL  TYPE ERROR ROUTINE
 6368                                   ;*ENTRY -       JSR     PC,TYPERR
 6369                                   ;*RETURN -      RTS     PC
 6370                                   ;*
 6371                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 6372                                   ;*ERROR IS TO BE REPORTED. IT THEN USES THE "ERROR TABLE" ($ERRTB)
 6373                                   ;*ENTRY TO DEFINE WHAT INFORMATION IS TO BE REPORTED CONCERNING
 6374                                   ;*THE ERROR.
 6375                                   ;;*****************************************************************
 6376  030330  104407                  TYPERR: SAVREG
 6377  030332  105237  003137                  INCB    DRVERS           ;INCR ERROR COUNT FOR THIS DRIVE
 6378  030336  032777  020000  150574  9$:     BIT     #SW13,@SWR       ;INHIBIT ERROR TYPEOUTS?
 6379  030344  001402                          BEQ     6$               ;BR IF NO
 6380  030346  000137  030752                  JMP     20$
 6381  030352  005000                  6$:     CLR     R0               ;CLR R0 FOR ERROR NUMBER
 6382  030354  005005                          CLR     R5               ;INIT INDENT INDICATOR
 6383  030356  005105                          COM     R5
 6384  030360  113700  001114                  MOVB    $ITEMB,R0        ;ENTER ERROR NUMBER
 6385  030364  005300                          DEC     R0               ;FORM INDEX FOR ERROR TABLE
 6386  030366  006300                          ASL     R0
 6387  030370  006300                          ASL     R0
 6388  030372  006300                          ASL     R0
 6389  030374  062700  001410          1$:     ADD     #$ERRTB,R0       ;FORM ADDRESS OF ERROR ENTRY
 6390  030400  012037  030420                  MOV     (R0)+,2$         ;GET EM POINTER
 6391  030404  001406                          BEQ     3$               ;BRANCH IF THERE ISN'T ONE
 6392  030406  104401  012103                  TYPE    ,CR2LF
 6393  030412  104401  045400                  TYPE    ,AS2SP2          ;TYPE "*  "
 6394  030416  104401                          TYPE                     ;TYPE ERROR MESSAGE (EM)
 6395  030420  000000                  2$:     .WORD   0                ;EM POINTER GOES HERE
 6396  030422  012037  030576          3$:     MOV     (R0)+,4$         ;GET DH POINTER
 6397  030426  001467                          BEQ     5$               ;BR IF THERE ISN'T ONE
 6398  030430  104401  001325                  TYPE    ,$CRLF
 6399  030434  104401  045366                  TYPE    ,TSTMSG          ;TYPE "    TEST "
 6400  030440  013746  001102                  MOV     $TSTNM,-(SP)     ;GET TEST NO. ON STACK
 6401  030444  104403                          TYPOS                    ;TYPE IT
 6402  030446     002                          .BYTE   2                ;2 DIGITS
 6403  030447     000                          .BYTE   0                ;SUPPRESS LEADING ZEROS
 6404  030450  104401  012103                  TYPE    ,CR2LF
 6405  030454  032777  010000  150456          BIT     #BIT12,@SWR      ;REPORT DESCRIPTION ONLY ?
 6406  030462  001133                          BNE     20$              ;BR IF YES
 6407  030464  104401  050522                  TYPE    ,DH10S           ;TYPE "PREVIOUS COMMAND :"
```

```
 6408  030470  104401  001325                    TYPE    ,$CRLF
 6409  030474  104401  050714                    TYPE    ,DH101+10       ;TYPE PREV COMMAND HEADER
 6410  030500  104401  001325                    TYPE    ,$CRLF
 6411  030504  012701  000006                    MOV     #6,R1           ;SIX COMMAND VALUES
 6412  030510  012702  001236                    MOV     #$REG26,R2      ;STARTING ADDR OF PREV CMND VALUES
 6413  030514  012246                  30$:      MOV     (R2)+,-(SP)     ;PUT A WORD ON STACK
 6414  030516  104402                            TYPOC                   ;TYPE IT
 6415  030520  104401  012117                    TYPE    ,SPACE2         ;TYPE SEPARATORS
 6416  030524  005301                            DEC     R1              ;SEE IF 7 VALUES TYPED YET
 6417  030526  001372                            BNE     30$             ;BR IF NOT
 6418  030530  104401  001325                    TYPE    ,$CRLF
 6419  030534  104401  012117                    TYPE    ,SPACE2         ;INDENT
 6420  030540  104401  050773                    TYPE    ,DH102          ;TYPE HDR FOR BA DATA
 6421  030544  104401  001325                    TYPE    ,$CRLF
 6422  030550  104401  012117                    TYPE    ,SPACE2         ;INDENT
 6423  030554  012246                            MOV     (R2)+,-(SP)
 6424  030556  104402                            TYPOC                   ;TYPE PREV. HI BA BITS
 6425  030560  104401  012117                    TYPE    ,SPACE2
 6426  030564  011246                            MOV     (R2),-(SP)
 6427  030566  104402                            TYPOC                   ;TYPE PREV. LO BA BITS
 6428  030570  104401  012103                    TYPE    ,CR2LF
 6429  030574  104401                            TYPE                    ;TYPE DATA HEADER
 6430  030576  000000                  4$:       .WORD   0               ;DH POINTER GOES HERE
 6431  030600  104401  001325                    TYPE    ,$CRLF
 6432  030604  005005                            CLR     R5              ;INIT INDENT INDICATOR
 6433  030606  032777  010000  150324  5$:       BIT     #BIT12,@SWR     ;REPORT DESCRIPTION ONLY ?
 6434  030614  001056                            BNE     20$             ;BR IF YES
 6435  030616  012001                            MOV     (R0)+,R1        ;GET DT POINTER
 6436  030620  001454                            BEQ     20$             ;BRANCH IF THERE ARE NONE
 6437  030622  012000                            MOV     (R0)+,R0        ;GET DF POINTER
 6438  030624  012002                            MOV     (R0)+,R2        ;STORE NUMBER OF DH'S
 6439  030626  112003                  10$:      MOVB    (R0)+,R3        ;GET & STORE NUMBER OF DATA WORDS
 6440  030630  105720                            TSTB    (R0)+           ;BUMP PAST FORMAT WORD
 6441  030632  005703                            TST     R3              ;TEST IF ANY DATA FOR THIS HEADER
 6442  030634  001417                            BEQ     14$             ;NO - SKIP DATA PRINT
 6443  030636  005705                            TST     R5              ;SEE IF SHOULD INDENT
 6444  030640  001002                            BNE     11$             ;BR IF NOT
 6445  030642  104401  012117                    TYPE    ,SPACE2         ;INDENT
 6446  030646  013146                  11$:      MOV     @(R1)+,-(SP)    ;PUT FIRST DATA WORD ON STACK
 6447  030650  104402                            TYPOC                   ;TYPE IT
 6448  030652  005303                            DEC     R3              ;MORE DATA WORDS
 6449  030654  001403                            BEQ     12$             ;NO-BRANCH
 6450  030656  104401  012117                    TYPE    ,SPACE2         ;TYPE SEPARATORS
 6451  030662  000771                            BR      11$             ;LOOP
 6452  030664  005702                  12$:      TST     R2              ;SEE IF <CR>,<LF> NEEDED
 6453  030666  001402                            BEQ     14$             ;BR IF NOT
 6454  030670  104401  001325                    TYPE    ,$CRLF          ;TYPE IT
 6455  030674  005302                  14$:      DEC     R2              ;MORE DH'S?
 6456  030676  003425                            BLE     20$             ;NO-BRANCH
 6457  030700  012037  030732          15$:      MOV     (R0)+,16$       ;GET NEXT DH POINTER
 6458  030704  105710                            TSTB    (R0)            ;SEE IF ANY DATA FOR HDR
 6459  030706  001004                            BNE     34$             ;BR IF YES
 6460  030710  104401  001325                    TYPE    ,$CRLF          ;SKIP EXTRA LINE
 6461  030714  005005                            CLR     R5              ;RE-INIT INDENT INDICATOR
 6462  030716  000404                            BR      36$
 6463  030720  005105                  34$:      COM     R5              ;COMPLEMENT INDENT INDICATOR
```

```
  6464  030722  001002                          BNE     36$             ;BR IF NO INDENT REQUIRED
  6465  030724  104401  012117                  TYPE    ,SPACE2         ;INDENT
  6466  030730  104401                  36$:    TYPE                    ;TYPE DH
  6467  030732  000000                  16$:    .WORD   0               ;DH POINTER GOES HERE
  6468  030734  104401  001325                  TYPE    ,$CRLF
  6469  030740  105710                          TSTB    (R0)            ;TYPE A DT?
  6470  030742  001331                          BNE     10$             ;YES-BRANCH
  6471  030744  062700  000002                  ADD     #2,R0           ;INCREMENT DF POINTER
  6472  030750  000751                          BR      14$             ;SEE IF END OF DF BLOCK
  6473  030752  104410                  20$:    RESREG
  6474  030754  000207                          RTS     PC
  6475                                  ;;*****************************************************************
  6476                                  .SBTTL  CONTROLLER ERROR REPORTER ROUTINE
  6477                                  ;*ENTRY:        JSR PC, CONERR
  6478                                  ;*RETURN:       RTS PC
  6479                                  ;*
  6480                                  ;*THIS ROUTINE DECODES THE CONTROLLER ERROR WORD AND
  6481                                  ;*REPORTS THE APPROPRIATE MESSAGE. THE RK611 REGISTERS ARE
  6482                                  ;*RETRIEVED FROM THE RK611 AND PLACED IN THE PARAMETER
  6483                                  ;*BLOCK. THIS IS DONE BECAUSE PARM 0 MAY NOT BE VALID
  6484                                  ;*AT THIS TIME.
  6485                                  ;;*****************************************************************
  6486  030756  104407                  CONERR: SAVREG                  ;SAVE R0-R5
  6487  030760  152737  000377  003132          BISB    #377,DONE       ;SET DONE FLAG
  6488  030766  105237  003135                  INCB    ERRCNT          ;INCREMENT ERROR COUNT
  6489  030772  004737  033752                  JSR     PC,TOPROC       ;LOAD RK REGS INTO $REGS
  6490  030776  032737  000001  003062          BIT     #BIT0,E.CONT    ;ERROR 0?
  6491  031004  001402                          BEQ     1$              ;NO-BRANCH
  6492  031006  104064                          ERROR   64              ;CLEAR CONT DID NOT CLEAR ERROR
  6493  031010  000474                          BR      7$
  6494  031012  032737  000002  003062  1$:     BIT     #BIT1,E.CONT    ;ERROR 1?
  6495  031020  001402                          BEQ     2$              ;NO-BRANCH
  6496  031022  104065                          ERROR   65              ;NO ATTENTION IN ATTENTION SUM REG
  6497  031024  000466                          BR      7$
  6498  031026  032737  000004  003062  2$:     BIT     #BIT2,E.CONT    ;ERROR 2?
  6499  031034  001407                          BEQ     3$              ;NO-BRANCH
  6500  031036  105737  003141                  TSTB    DRNAFG          ;SEE IF DRIVE WAS SIEZED BY OTHER PORT
  6501  031042  001402                          BEQ     15$             ;BR IF NOT
  6502  031044  105237  003142                  INCB    REISSU          ;SET FLAG TO RE-ISSUE COMMAND
  6503  031050  104066                  15$:    ERROR   66              ;UNSOLICITED ATTENTION
  6504  031052  000453                          BR      7$
  6505  031054  032737  000010  003062  3$:     BIT     #BIT3,E.CONT    ;ERROR 3?
  6506  031062  001402                          BEQ     4$              ;NO-BRANCH
  6507  031064  104067                          ERROR   67              ;UNEXPECTED DATA TYPE ERROR
  6508  031066  000445                          BR      7$
  6509  031070  032737  000020  003062  4$:     BIT     #BIT4,E.CONT    ;ERROR 4?
  6510  031076  001402                          BEQ     5$              ;NO-BRANCH
  6511  031100  104070                          ERROR   70              ;ATTENTION DID NOT RESET WITH CLEAR
  6512  031102  000437                          BR      7$
  6513  031104  032737  000040  003062  5$:     BIT     #BIT5,E.CONT    ;ERROR 5?
  6514  031112  001402                          BEQ     6$              ;NO-BRANCH
  6515  031114  104071                          ERROR   71              ;SUBSYS CLEAR DIDN'T CLEAR DRIVE ATTENTION
  6516  031116  000431                          BR      7$
  6517  031120  032737  000400  003062  6$:     BIT     #BIT8,E.CONT
  6518  031126  001402                          BEQ     8$
  6519  031130  104072                          ERROR   72              ;DATA LATE WHEN UNLOADING HEADER
```

```
 6520  031132  000423                          BR      7$
 6521  031134  032737  001000  003062  8$:     BIT     #BIT9,E.CONT
 6522  031142  001402                          BEQ     9$
 6523  031144  104073                          ERROR   73              ;CONTROLLER ERROR DURING DRIVER SERVICE
 6524  031146  000415                          BR      7$
 6525  031150  032737  002000  003062  9$:     BIT     #BIT10,E.CONT
 6526  031156  001402                          BEQ     10$
 6527  031160  104074                          ERROR   74              ;DRIVE DETECTED PARITY ERROR
 6528  031162  000407                          BR      7$
 6529  031164  032737  100000  003062  10$:    BIT     #BIT15,E.CONT
 6530  031172  001402                          BEQ     11$
 6531  031174  104052                          ERROR   52              ;MULTIPLE DRIVE SELECT
 6532  031176  000401                          BR      7$
 6533  031200  104075                  11$:    ERROR   75              ;UNDEFINED ERROR
 6534  031202  005037  003062          7$:     CLR     E.CONT          ;CLEAR CONTROLLER ERROR WORD
 6535  031206  000137  033546                  JMP     BGNRTY          ;GO DO RETRY
 6536                                  ;;*****************************************************************
 6537                                  .SBTTL  REPORT SUPPORT ROUTINE
 6538                                  ;*THIS ROUTINE LOADS ALL THE PARAMETER BLOCK DATA TO BE REPORTED
 6539                                  ;*INTO THE PROPER TEMPORARY REGISTERS FOR REPORTING.  ALL THESE MAY
 6540                                  ;*NOT BE INCLUDED IN THE REPORT, BUT THEY ARE LOADED ANYWAY.
 6541  031212                          REPSUP:
 6542  031212  104407                          SAVREG
 6543  031214  005037  005426                  CLR     ERRCOM          ;CLEAR ERROR COMMAND STORE
 6544  031220  116537  000001  005426          MOVB    P.CMND(R5),ERRCOM ;STORE COMMAND START VALUES
 6545  031226  012700  001162                  MOV     #$REG0,R0       ;FOR REPORTING
 6546  031232  016520  000002                  MOV     P.CYLN(R5),(R0)+
 6547  031236  116520  000005                  MOVB    P.TRCK(R5),(R0)+
 6548  031242  105020                          CLRB    (R0)+
 6549  031244  116520  000004                  MOVB    P.SECT(R5),(R0)+
 6550  031250  105020                          CLRB    (R0)+
 6551  031252  016520  000012                  MOV     P.WC(R5),(R0)+
 6552  031256  012700  001174                  MOV     #$REG5,R0
 6553  031262  116503  000007                  MOVB    P.BAHI(R5),R3
 6554  031266  042703  177774                  BIC     #177774,R3
 6555  031272  010337  001256                  MOV     R3,$REG36       ;HI BA BITS
 6556  031276  016537  000010  001260          MOV     P.BALO(R5),$REG37 ;LO BA BITS
 6557  031304  016520  000016                  MOV     P.CS1(R5),(R0)+ ;GET ALL THE VALUES FROM THE
 6558  031310  016520  000020                  MOV     P.CS2(R5),(R0)+ ;PARAMETER BLOCK AND LOAD
 6559  031314  016520  000030                  MOV     P.DCYL(R5),(R0)+ ;THE TEMPORARY REGISTERS
 6560  031320  016520  000026                  MOV     P.DTS(R5),(R0)+ ;FOR REPORTING. ALL THIS
 6561  031324  016520  000022                  MOV     P.WCR(R5),(R0)+ ;DATA MAY NOT BE VALID
 6562                                                                  ;FOR ALL REPORTS (TO BE
 6563  031330  016520  000024                  MOV     P.BAR(R5),(R0)+ ;DETERMINED LATER) BUT IT IS
 6564  031334  016520  000032                  MOV     P.ASOF(R5),(R0)+ ;STORED ANY WAY.
 6565  031340  016520  000036                  MOV     P.DS(R5),(R0)+
 6566  031344  016520  000034                  MOV     P.ER(R5),(R0)+
 6567  031350  016520  000040                  MOV     P.A00(R5),(R0)+
 6568  031354  016520  000042                  MOV     P.B00(R5),(R0)+
 6569  031360  016520  000044                  MOV     P.A01(R5),(R0)+
 6570  031364  016520  000046                  MOV     P.B01(R5),(R0)+
 6571  031370  016520  000050                  MOV     P.A10(R5),(R0)+
 6572  031374  016520  000052                  MOV     P.B10(R5),(R0)+
 6573  031400  016520  000054                  MOV     P.A11(R5),(R0)+
 6574  031404  016520  000056                  MOV     P.B11(R5),(R0)+
 6575                                  ;STORE PREVIOUS COMMAND FOR PRINTOUT
```

```
  6576  031410  012701  001236                    MOV     #$REG26,R1      ;ADRS OF PRINT BUF AREA
  6577  031414  012700  005162                    MOV     #PRVCMD,R0      ;ADRS OF PREV CMND STORAGE
  6578  031420  112021                            MOVB    (R0)+,(R1)+     ;DRIVE NO.
  6579  031422  105021                            CLRB    (R1)+
  6580  031424  112021                            MOVB    (R0)+,(R1)+     ;COMMAND
  6581  031426  105021                            CLRB    (R1)+
  6582  031430  012021                            MOV     (R0)+,(R1)+     ;CYL ADDRESS
  6583  031432  116021  000001                    MOVB    1(R0),(R1)+     ;TRACK
  6584  031436  105021                            CLRB    (R1)+
  6585  031440  111021                            MOVB    (R0),(R1)+      ;SECTOR
  6586  031442  105021                            CLRB    (R1)+
  6587  031444  016021  000006                    MOV     6(R0),(R1)+     ;WORD COUNT
  6588  031450  116003  000003                    MOVB    3(R0),R3        ;HI BA BITS
  6589  031454  042703  177774                    BIC     #177774,R3
  6590  031460  010321                            MOV     R3,(R1)+
  6591  031462  016011  000004                    MOV     4(R0),(R1)      ;LO BA BITS
  6592  031466  104410                            RESREG
  6593  031470  000207                            RTS     PC
  6594                          ;;*****************************************************************
  6595                          .SBTTL  REPORT ERROR ROUTINE
  6596                          ;* ENTRY         JSR     PC, ERRHDL
  6597                          ;*RETURN        RTS     PC
  6598                          ;*
  6599                          ;*THIS ROUTINE IS CALLED BY THE DRIVER WHEN AN ERROR IS DETECTED
  6600                          ;*IN THE OPERATION.  THE ROUTINE DETERMINES WHICH COMMAND WAS
  6601                          ;*BEING EXECUTED AND GENERATES THE PROPER ERROR MESSAGE.
  6602                          ;;*****************************************************************
  6603
  6604  031472  104407                    ERRHDL: SAVREG
  6605  031474  152737  000377  003132            BISB    #377,DONE       ;SET DONE FLAG
  6606  031502  105237  003135                    INCB    ERRCNT          ;INCREMENT ERROR COUNT
  6607  031506  005037  005424                    CLR     RECODE          ;CLEAR RECOVERY CODE WORD
  6608  031512  032737  000400  005424  ER2ENT:   BIT     #LEV2ER,RECODE  ;TEST IF 2ND LEVEL ERROR
  6609  031520  001402                            BEQ     52$             ;NO - SKIP PARAM BLOCK CHANGE
  6610  031522  012705  002724                    MOV     #PARM1,R5       ;ELSE SET R5 TO PARAMETER BLOCK 1
  6611  031526  012737  033724  003056  52$:     MOV     #RETANL,A.ABNL  ;SET NOW ABNORMAL AND NORMAL RETURN FOR
  6612  031534  012737  033742  003054            MOV     #RETNML,A.NORM  ;DRIVER OPERATIONS IN ERROR PROCESSING
  6613  031542  004737  031212                    JSR     PC,REPSUP       ;GO SET UP REGISTERS FOR REPORT
  6614                                                                    ;NOW BEGIN TESTING THE ERROR
  6615                                                                    ;BITS. THE SEQUENCE IN
  6616                                                                    ;WHICH THEY ARE TESTED IS
  6617                                                                    ;CONSIDERED SIGNIFICANT IN
  6618                                                                    ;THAT ERRORS OF A MORE
  6619                                                                    ;CATASTROPHIC NATURE ARE FIRST
  6620                                                                    ;TESTED.
  6621                                                                    ;
  6622                                                                    ;IF AN ERROR IS FOUND SET,
  6623                                                                    ;THAT ERROR IS REPORTED AND
  6624                                                                    ;THE REPORTING IS TERMINATED.
  6625                                                                    ;IF ADDITIONAL ERRORS ARE SET,
  6626                                                                    ;THE RK611 REGISTER PRINTOUTS
  6627                                                                    ;WILL SHOW THIS BUT THE
  6628                                                                    ;REGISTER CONTENTS MUST BE
  6629                                                                    ;MANUALLY DECODED TO LOCATE THE
  6630                                                                    ;SECOND ERROR
  6631  031546  016504  000034                    MOV     P.ER(R5),R4     ;SET R4 TO ERROR REGISTER
```

```
 6632  031552  032765  020000  000020          BIT     #UPE,P.CS2(R5)  ;TEST UPE. IF UES-SET
 6633  031560  001406                          BEQ     1$              ;REPORT ERROR
 6634  031562  104001                          ERROR   1
 6635  031564  052737  000200  005424          BIS     #ABORT,RECODE
 6636  031572  000137  033224                  JMP     37$
 6637  031576  032765  004000  000020  1$:     BIT     #NEM,P.CS2(R5)  ;TEST NON-EXISTANT MEMORY
 6638  031604  001406                          BEQ     2$
 6639  031606  104002                          ERROR   2
 6640  031610  052737  000200  005424          BIS     #ABORT,RECODE
 6641  031616  000137  033224                  JMP     37$
 6642  031622  032765  010000  000020  2$:     BIT     #NED,P.CS2(R5)  ;TEST NON-EXISTANT DRIVE
 6643  031630  001412                          BEQ     3$
 6644  031632  032765  000400  000020          BIT     #UFE,P.CS2(R5)  ;TEST IF NED & UFE BOTH SET
 6645  031640  001403                          BEQ     38$
 6646  031642  104035                          ERROR   35              ;NED & UFE BOTH SET ERROR
 6647  031644  000137  033224                  JMP     37$
 6648  031650  104003                  38$:    ERROR   3               ;NED ONLY
 6649  031652  000137  033224                  JMP     37$
 6650  031656  032765  000400  000020  3$:     BIT     #UFE,P.CS2(R5)  ;TEST UNIT FIELD ERROR
 6651  031664  001412                          BEQ     4$
 6652  031666  032765  010000  000020          BIT     #NED,P.CS2(R5)  ;TEST IF UFE & NED BOTH SET
 6653  031674  001403                          BEQ     39$             ;NO-SKIP
 6654  031676  104035                          ERROR   35              ;REPORT NED & UFE BOTH SET
 6655  031700  000137  033224                  JMP     37$
 6656  031704  104004                  39$:    ERROR   4               ;REPORT UFE ONLY
 6657  031706  000137  033224                  JMP     37$
 6658  031712  032765  000100  000014  4$:     BIT     #CMDTO,P.PRST(R5) ;
 6659  031720  001423                          BEQ     5$
 6660  031722  004737  033752                  JSR     PC,TOPROC       ;GO PROCESS TIMEOUT
 6661  031726  032737  002000  005424          BIT     #TWOTOS,RECODE  ;2ND TIMEOUT IN TIMEOUT PROC?
 6662  031734  001007                          BNE     40$             ;YES - SKIP TO ERROR 56
 6663  031736  032737  000400  005424          BIT     #LEV2ER,RECODE  ;TEST IF LEVEL 2 ERROR
 6664  031744  001006                          BNE     41$             ;YES - SKIP TO ERROR 57
 6665  031746  104005                          ERROR   5               ;ELSE MAKE FULL TIMEOUT REPORT
 6666  031750  000137  033224                  JMP     37$
 6667  031754  104056                  40$:    ERROR   56              ;TWO TIMEOUTS ERROR REPORT
 6668  031756  000137  033224                  JMP     37$
 6669  031762  104057                  41$:    ERROR   57              ;2ND LEVEL ERROR REPORT
 6670  031764  000137  031512                  JMP     ER2ENT          ;GO BUILD AND MAKE 2ND REPORT
 6671  031770  032765  010000  000014  5$:     BIT     #DRVSZD,P.PRST(R5) ;SEE IF DRIVE SIEZED BY OTHER PORT
 6672  031776  001403                          BEQ     65$             ;BR IF DRIVE IS AVAILABLE
 6673  032000  104107                          ERROR   107             ;DRIVE SIEZED BY OTHER PORT
 6674  032002  000137  033224                  JMP     37$
 6675  032006  032765  020000  000016  65$:    BIT     #SPAR,P.CS1(R5) ;TEST D TO C PARITY ERROR
 6676  032014  001406                          BEQ     6$
 6677  032016  104006                          ERROR   6
 6678  032020  052737  004000  005424          BIS     #RCLREQ,RECODE
 6679  032026  000137  033224                  JMP     37$
 6680  032032  032704  000010          6$:     BIT     #DRPAR,R4       ;TEST DRIVE DETECTED PARITY ERROR
 6681  032036  001406                          BEQ     7$
 6682  032040  104007                          ERROR   7
 6683  032042  052737  004000  005424          BIS     #RCLREQ,RECODE
 6684  032050  000137  033224                  JMP     37$
 6685  032054  032765  000010  000036  7$:     BIT     #ACLO,P.DS(R5)  ;TEST AC LOW
 6686  032062  001403                          BEQ     8$
 6687  032064  104010                          ERROR   10
```

```
 6688  032066  000137  033224                    JMP     37$
 6689  032072  032765  000020  000036  8$:       BIT     #SPDLSS,P.DS(R5) ;TEST SPEED LOSS
 6690  032100  001403                            BEQ     24$
 6691  032102  104011                            ERROR   11
 6692  032104  000137  033224                    JMP     37$
 6693  032110  032765  004000  000016  24$:      BIT     #CTO,P.CS1(R5)  ;TEST FOR CONTROLLER TIMEOUT
 6694  032116  001401                            BEQ     25$
 6695  032120  104027                            ERROR   27
 6696  032122  032704  000001          25$:      BIT     #ILC,R4         ;TEST ILLEGAL FUNCTION CODE
 6697  032126  001403                            BEQ     10$
 6698  032130  104012                            ERROR   12
 6699  032132  000137  033224                    JMP     37$
 6700  032136  032765  002000  000020  10$:      BIT     #PGE,P.CS2(R5)  ;TEST PROGRAMMING ERROR
 6701  032144  001403                            BEQ     11$
 6702  032146  104013                            ERROR   13
 6703  032150  000137  033224                    JMP     37$
 6704  032154  032704  000004          11$:      BIT     #ILF,R4         ;TEST ILLEGAL DRIVE FUNCTION
 6705  032160  001403                            BEQ     12$
 6706  032162  104014                            ERROR   14
 6707  032164  000137  033224                    JMP     37$
 6708  032170  032704  000040          12$:      BIT     #DTYE,R4        ;TEST DRIVE TYPE ERROR
 6709  032174  001403                            BEQ     13$
 6710  032176  104015                            ERROR   15
 6711  032200  000137  033224                    JMP     37$
 6712  032204  032704  000020          13$:      BIT     #FMTE,R4        ;TEST FORMAT ERROR
 6713  032210  001403                            BEQ     14$
 6714  032212  104016                            ERROR   16
 6715  032214  000137  033224                    JMP     37$
 6716  032220  032704  004000          14$:      BIT     #WLE,R4         ;TEST WRITE LOCK ERROR
 6717  032224  001403                            BEQ     15$
 6718  032226  104017                            ERROR   17
 6719  032230  000137  033224                    JMP     37$
 6720  032234  032704  040000          15$:      BIT     #UNS,R4         ;TEST DRIVE UNSAFE
 6721  032240  001406                            BEQ     16$
 6722  032242  104020                            ERROR   20
 6723  032244  052737  000200  005424            BIS     #ABORT,RECODE
 6724  032252  000137  033324                    JMP     ALLTRM
 6725  032256  032704  000002          16$:      BIT     #SKI,R4         ;TEST SEEK INCOMPLETE
 6726  032262  001406                            BEQ     17$
 6727  032264  104021                            ERROR   21
 6728  032266  052737  004000  005424            BIS     #RCLREQ,RECODE
 6729  032274  000137  033224                    JMP     37$
 6730  032300  032704  001000          17$:      BIT     #COE,R4         ;TEST CYLINDER OVERFLOW
 6731  032304  001406                            BEQ     18$
 6732  032306  104022                            ERROR   22
 6733  032310  052737  004000  005424            BIS     #RCLREQ,RECODE
 6734  032316  000137  033224                    JMP     37$
 6735  032322  032704  002000          18$:      BIT     #IDAE,R4        ;TEST ILLEGAL CYLINDER
 6736  032326  001406                            BEQ     19$
 6737  032330  104023                            ERROR   23
 6738  032332  052737  004000  005424            BIS     #RCLREQ,RECODE
 6739  032340  000137  033224                    JMP     37$
 6740  032344  032765  000040  000036  19$:      BIT     #DROT,P.DS(R5)  ;TEST DRIVE OFF TRACK
 6741  032352  001406                            BEQ     20$
 6742  032354  104024                            ERROR   24
 6743  032356  052737  004000  005424            BIS     #RCLREQ,RECODE
```

```
 6744  032364  000137  033224                    JMP     37$
 6745  032370  032704  010000          20$:      BIT     #DTE,R4          ;TEST DRIVE TIMING ERROR
 6746  032374  001406                            BEQ     21$
 6747  032376  104025                            ERROR   25
 6748  032400  052737  000200  005424            BIS     #ABORT,RECODE
 6749  032406  000137  033224                    JMP     37$
 6750  032412  032765  100000  000020  21$:      BIT     #DLT,P.CS2(R5)   ;TEST DATA LATE
 6751  032420  001403                            BEQ     22$
 6752  032422  104026                            ERROR   26
 6753  032424  000137  033224                    JMP     37$
 6754  032430  032704  020000          22$:      BIT     #OPI,R4          ;TEST IF OPI ERROR
 6755  032434  001470                            BEQ     29$
 6756  032436  052737  000010  005424            BIS     #OPIERR,RECODE
 6757  032444  105737  003130                    TSTB    DERCNT           ;TEST IF FIRST ERROR
 6758  032450  001402                            BEQ     50$
 6759  032452  000137  033224                    JMP     37$              ;NO - SKIP REPORT
 6760  032456  032737  000400  005424  50$:      BIT     #LEV2ER,RECODE   ;TEST IF A SECOND LEVEL 2 ERROR
 6761  032464  001403                            BEQ     26$              ;HAS ALREADY OCCURRED
 6762  032466  104054                  27$:      ERROR   54
 6763  032470  000137  033224                    JMP     37$              ;GET OUT OF ERROR REPORT
 6764  032474  004737  034432          26$:      JSR     PC,BLDEXH        ;GO BUILD EXPECTED HEADER
 6765  032500  004737  034122                    JSR     PC,RDHD0         ;GET HEADER OF SECTOR 0
 6766  032504  032737  000400  005424            BIT     #LEV2ER,RECODE   ;TEST IF ERROR GETTING HDR
 6767  032512  001036                            BNE     28$              ;IF YES-GO MAKE ABREVIATED REPORT
 6768  032514  013702  003046                    MOV     RKBAS,R2         ;STORE HEADER 0 INTO REGISTERS
 6769  032520  042762  000100  000000            BIC     #IE,RKCS1(R2)    ;RESET INTERRUPT ENABLE
 6770  032526  016237  000024  001202            MOV     RKDB(R2),$REG10  ;FOR REPORTING
 6771  032534  016237  000024  001204            MOV     RKDB(R2),$REG11
 6772  032542  016237  000024  001206            MOV     RKDB(R2),$REG12
 6773  032550  032762  100000  000000            BIT     #CERR,RKCS1(R2)  ;TEST IF ERROR DURING STORAGE
 6774  032556  001343                            BNE     27$
 6775  032560  105737  003143                    TSTB    TSTTYP           ;SEE IF READING WITH OFFSET
 6776  032564  001403                            BEQ     54$              ;BR IF NOT
 6777  032566  105037  003135                    CLRB    ERRCNT           ;CLEAR ERROR COUNT
 6778  032572  000401                            BR      55$              ;CONTINUE
 6779  032574  104030                  54$:      ERROR   30               ;MAKE OPI REPORT
 6780  032576  052737  004000  005424  55$:      BIS     #RCLREQ,RECODE   ;SET RECALIBRATE REQUEST BIT
 6781  032604  000137  033224                    JMP     37$
 6782  032610  104054                  28$:      ERROR   54
 6783  032612  000137  031512                    JMP     ER2ENT           ;GO MAKE 2ND LEVEL REPORT
 6784  032616  032704  000400          29$:      BIT     #HVRC,R4         ;TEST IF HVRC ERROR
 6785  032622  001457                            BEQ     23$
 6786  032624  052737  000004  005424            BIS     #HVRCER,RECODE
 6787  032632  105737  003130                    TSTB    DERCNT           ;TEST IF FIRST ERROR
 6788  032636  001402                            BEQ     30$              ;YES - REPORT
 6789  032640  000137  033224                    JMP     37$              ;JUMP TO RETURN
 6790  032644  032737  000400  005424  30$:      BIT     #LEV2ER,RECODE   ;TEST IF A 2ND LEVEL ERROR HAS ALREADY
 6791  032652  001403                            BEQ     31$              ;OCCURRED. NO-SKIP EXIT
 6792  032654  104055                  51$:      ERROR   55
 6793  032656  000137  033224                    JMP     37$              ;GET OUT OF ERROR REPORT
 6794  032662  004737  034432          31$:      JSR     PC,BLDEXH        ;GO BUILD EXPECTED HEADER
 6795  032666  004737  034122                    JSR     PC,RDHD0         ;GO GET HDR 0
 6796  032672  032737  000400  005424            BIT     #LEV2ER,RECODE   ;TEST IF ERROR IN GETTING HDR
 6797  032700  001025                            BNE     32$              ;IF YES-GO MAKE ABREVIATED REPORT
 6798  032702  013702  003046                    MOV     RKBAS,R2         ;GET RK611 BASE ADDRESS
 6799  032706  042762  000100  000000            BIC     #IE,RKCS1(R2)    ;CLEAR INTERRUPT ENABLE
```

```
 6800  032714  016237  000024  001202          MOV     RKDB(R2),$REG10 ;STORE OFF HEADER
 6801  032722  016237  000024  001204          MOV     RKDB(R2),$REG11
 6802  032730  016237  000024  001206          MOV     RKDB(R2),$REG12
 6803  032736  032762  100000  000000          BIT     #CERR,RKCS1(R2) ;TEST IN ANY ERROR IN UNLOAD
 6804  032744  001343                          BNE     51$             ;IY YES - GO MAKE SHORT REPORT
 6805  032746  104031                          ERROR   31              ;MAKE FULL REPORT
 6806  032750  000137  033224                  JMP     37$
 6807  032754  104055                  32$:    ERROR   55              ;ABREVIATED HVRC ERROR RPORT
 6808  032756  000137  031512                  JMP     ER2ENT          ;GO REPORT 2ND LEVEL ERROR
 6809  032762  032704  000200          23$:    BIT     #BSE,R4         ;TEST FOR BAD SECTOR ERROR
 6810  032766  001436                          BEQ     33$             ;NO - SKIP
 6811  032770  052737  000002  005424          BIS     #BSERR,RECODE   ;SET ERROR FLAG
 6812  032776  016500  000030                  MOV     P.DCYL(R5),R0   ;GET CYL NO.
 6813  033002  116501  000027                  MOVB    P.DTS+1(R5),R1  ;GET TRACK NO.
 6814  033006  042701  177774                  BIC     #177774,R1      ;CLEAR ALL BITS EXCEPT TRACK
 6815  033012  016502  000026                  MOV     P.DTS(R5),R2    ;GET SECTOR IN ERROR
 6816  033016  042702  177740                  BIC     #177740,R2      ;CLEAR ALL BITS EXCEPT SECTOR
 6817  033022  004737  030020                  JSR     PC,BDSRCK       ;GO SEE IF THIS SECTOR LISTED
 6818  033026  032737  001000  005424          BIT     #BADSEC,RECODE  ;TEST RESULT
 6819  033034  001073                          BNE     37$             ;YES - EXIT, NO ERROR OR REPORT
 6820  033036  105737  003143                  TSTB    TSTTYP          ;SEE IF READING WITH OFFSET
 6821  033042  001403                          BEQ     56$             ;BR IF NOT
 6822  033044  105037  003135                  CLRB    ERRCNT          ;CLEAR ERROR COUNT
 6823  033050  000465                          BR      37$             ;CONTINUE
 6824  033052  104104                  56$:    ERROR   104             ;ELSE REPORT BAD BSE
 6825  033054  042737  000002  005424          BIC     #BSERR,RECODE   ;RESET BSE ERROR FLAG
 6826  033062  000460                          BR      37$             ;GO EXIT
 6827  033064  032704  100000          33$:    BIT     #DCK,R4         ;TEST IF DATA CHECK
 6828  033070  001435                          BEQ     36$
 6829  033072  052737  000020  005424          BIS     #DCKERR,RECODE  ;SET DATA CHECK ERROR IN RECOVERY CODE
 6830  033100  032704  000100                  BIT     #ECH,R4         ;TEST IF ECC IS HARD. IF
 6831  033104  001406                          BEQ     34$             ;YES SET UNCORRECTABLE IN
 6832  033106  052737  000040  005424          BIS     #ECCNC,RECODE   ;RECOVERY FLAG AND A 0 IN
 6833  033114  005037  001206                  CLR     $REG12          ;REG12 TO INDICATE UNCORRECTABLE,
 6834  033120  000403                          BR      35$             ;A 1 IN REG 12 FOR CORRECTABLE
 6835  033122  012737  000001  001206  34$:    MOV     #1,$REG12
 6836  033130  105737  003130          35$:    TSTB    DERCNT          ;TEST IF FIRST ERROR
 6837  033134  001033                          BNE     37$             ;NO SKIP REPORT
 6838  033136  004737  034276                  JSR     PC,GTPKAD       ;GO GET PACK ADDRESS OF ERROR
 6839  033142  016537  000060  001202          MOV     P.EPOS(R5),$REG10 ;STORE ECC POSITION &
 6840  033150  016537  000062  001204          MOV     P.EPAT(R5),$REG11 ;PATTERN
 6841  033156  104032                          ERROR   32              ;REPORT DCK ERROR
 6842  033160  000137  033224                  JMP     37$
 6843  033164  032765  040000  000020  36$:    BIT     #WCE,P.CS2(R5)  ;TEST WRITE CHECK ERROR
 6844  033172  001414                          BEQ     37$
 6845  033174  042737  000200  005424          BIC     #ABORT,RECODE   ;CLEAR ABORT & SET WRITE
 6846  033202  052737  000100  005424          BIS     #WCERR,RECODE   ;CHECK ERROR IN RECODE
 6847  033210  105737  003130                  TSTB    DERCNT          ;TEST IF FIRST ERROR
 6848  033214  001003                          BNE     37$             ;NO - SKIP
 6849  033216  004737  034276                  JSR     PC,GTPKAD       ;GO GET ADDRESS OF ERROR
 6850  033222  104033                          ERROR   33              ;REPORT WCE
 6851
 6852  033224  032765  000020  000014  37$:    BIT     #DRVHRD,P.PRST(R5) ;TEST HARD ERROR
 6853  033232  001404                          BEQ     43$
 6854  033234  104036                          ERROR   36
 6855  033236  052737  000200  005424          BIS     #ABORT,RECODE
```

```
 6856
 6857  033244  032765  000040  000014  43$:    BIT     #DRVDSC,P.PRST(R5) ;TEST STATUS CHANGE NOT CLEARED
 6858  033252  001404                          BEQ     44$
 6859  033254  104037                          ERROR   37
 6860  033256  052737  000200  005424          BIS     #ABORT,RECODE
 6861
 6862  033264  032765  004000  000014  44$:    BIT     #NODSC,P.PRST(R5) ;IFST ATTENTION BUT NO FAULT OR DSC
 6863  033272  001404                          BEQ     46$
 6864  033274  104040                          ERROR   40
 6865  033276  052737  000200  005424          BIS     #ABORT,RECODE
 6866
 6867  033304  032765  000010  000014  46$:    BIT     #UEXATT,P.PRST(R5) ;TEST UNEXPECTED ATTENTION
 6868  033312  001404                          BEQ     ALLTRM
 6869  033314  104042                          ERROR   42
 6870  033316  052737  000200  005424          BIS     #ABORT,RECODE
 6871
 6872
 6873                                  ;ALL ERRORS MUST EXIT THROUGH THIS POINT
 6874
 6875  033324  012705  002640          ALLTRM: MOV     #PARM0,R5       ;RESTORE PARAMETER BLOCK SELECTION
 6876  033330  012737  020526  003056          MOV     #DCKHDL,A.ABNL  ;SET READ ERROR HANDLER
 6877  033336  105737  003143                  TSTB    TSTTYP          ;SEE IF READING WITH OFFSETS
 6878  033342  001003                          BNE     20$             ;BR IF YES
 6879  033344  012737  031472  003056          MOV     #ERRHDL,A.ABNL  ;RESTORE INTERRUPT VECTORS FOR RETRY
 6880  033352  012737  030250  003054  20$:    MOV     #ERRFRE,A.NORM  ;DRIVER OPERATIONS, IF ANY
 6881  033360  032737  000200  005424          BIT     #ABORT,RECODE   ;IF ABORT IS NOT SET AND
 6882  033366  001057                          BNE     48$             ;THE DRIVE IS READY (HAS NOT
 6883                                                                  ;CYCLED DOWN)
 6884  033370  013702  003046                  MOV     RKBAS,R2        ;GET BASE ADDRESS
 6885  033374  032762  000200  000012          BIT     #RDY,RKDS(R2)   ;TEST IF DRIVE READY SET
 6886  033402  001004                          BNE     47$             ;RECALIBRATE REQUIRED BIT IS SET
 6887  033404  052737  000200  005424          BIS     #ABORT,RECODE   ;ELSE ABORT WITH ABORT MESSAGE
 6888  033412  000445                          BR      48$
 6889  033414  032737  004000  005424  47$:    BIT     #RCLREQ,RECODE  ;IF RECALIBRATE IS REQUIRED
 6890  033422  001451                          BEQ     BGNRTY          ;FOR RETRY SET UP PARAM
 6891  033424  013737  002640  002724          MOV     PARM0,PARM1     ;SAVE COMMAND
 6892  033432  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;BLOCK TO DO IT.
 6893  033440  012737  033724  003056          MOV     #RETANL,A.ABNL
 6894  033446  004737  030124                  JSR     PC,DRVCAL
 6895  033452  013737  002724  002640          MOV     PARM1,PARM0     ;RESTORE COMMAND
 6896  033460  012737  020526  003056          MOV     #DCKHDL,A.ABNL  ;SET READ ERROR HANDLER
 6897  033466  105737  003143                  TSTB    TSTTYP          ;SEE IF READING WITH OFFSETS
 6898  033472  001003                          BNE     50$             ;BR IF YES
 6899  033474  012737  031472  003056          MOV     #ERRHDL,A.ABNL  ;RESTORE ERROR RETURN
 6900  033502  032737  000400  005424  50$:    BIT     #LEV2ER,RECODE  ;IF AN ERROR OCCURRED IN THE
 6901  033510  001416                          BEQ     BGNRTY          ;RECAL ATTEMP SET ABORT,
 6902  033512  052737  000200  005424          BIS     #ABORT,RECODE   ;PRINT THE RECAL ERROR MESSAGE, AND
 6903  033520  104060                          ERROR   60              ;GO REPORT DETAILS
 6904  033522  000137  031512                  JMP     ER2ENT
 6905  033526  104061                  48$:    ERROR   61              ;REPORT ABORT-RETRY FAILED
 6906
 6907                                  ;THE PROGRAM WILL HALT HERE IF THE ABORT FLAG HAS BEEN SET OR
 6908                                  ;IF THE DRIVE READY BIT IS RESET.
 6909
 6910  033530  000005                  HLTPRG: RESET                   ;DISABLE ALL DEVICES
 6911  033532  000000                          HALT                    ;HALT THE CPU
```

```
 6912  033534  105037  003135                  CLRB    ERRCNT          ;CLEAR ERROR COUNT
 6913  033540  000005                          RESET                   ;RESET ALL DEVICES
 6914  033542  000137  012206                  JMP     CMSTRT
 6915
 6916
 6917                                  ;THE FOLLOWING CODE WILL DETERMINE IF AND DATA TYPE ERROR
 6918                                  ;HAS OCCURRED. IF YES, THE RETRY IS NOT DONE HERE BUT RETURNS TO
 6919                                  ; THE INITIATING ROUTINE FOR RETRY.  ANY OTHER ERROR IS TO BE
 6920                                  ;RETRIED HERE. IF RETRY IS UNSUCCESSFUL AFTER 4 ATTEMPTS, THE ABORT
 6921                                  ;FLAG IS SET AND PROGRAM HALTS.
 6922
 6923  033546  032737  000136  005424  BGNRTY: BIT     #BSERR!HVRCER!OPIERR!DCKERR!WCERR,RECODE ;TEST IF ANY DATA ERROR
 6924  033554  001404                          BEQ     3$
 6925  033556  052737  100000  005424  9$:     BIS     #ANYDER,RECODE
 6926  033564  000453                          BR      2$
 6927  033566                          3$:
 6928  033566  105737  003150                  TSTB    NORTRY          ;SEE IF "NO-RETRY" FLAG SET
 6929  033572  001371                          BNE     9$              ;BR IF YES
 6930  033574  032737  001000  005424          BIT     #BADSEC,RECODE  ;TEST IF BAD SECTOR FLAG SET
 6931  033602  001044                          BNE     2$              ;IF YES-EXIT TO CALLER
 6932  033604  123737  003135  003136          CMPB    ERRCNT,ERRLMT   ;BEGIN RETRY IF ERROR COUNT HAS
 6933  033612  001012                          BNE     1$              ;NOT BEEN EXCEEDED
 6934  033614  005037  001174                  CLR     $REG5
 6935  033620  113737  003135  001174          MOVB    ERRCNT,$REG5    ;GET ERROR RETRY COUNT
 6936  033626  104102                          ERROR   102             ;REPORT RETRY UNSUCCESSFUL
 6937  033630  052737  000200  005424          BIS     #ABORT,RECODE   ;SET ABORT & QUIT
 6938  033636  000734                          BR      HLTPRG
 6939  033640  013702  003046          1$:     MOV     RKBAS,R2        ;GET RK BASE ADDRESS
 6940  033644  112765  000177  000001          MOVB    #SUBCLR,P.CMND(R5)  ;SET UP TO CLEAR SUBSYSTEM
 6941  033652  004737  030124                  JSR     PC,DRVCAL       ;GO DO IT
 6942  033656  012700  005176                  MOV     #COMSTR,R0      ;GO AND REESTABLISH THE COMMAND
 6943  033662  012025                          MOV     (R0)+,(R5)+     ;AS IT WAS ENTERED INTO THE
 6944  033664  012025                          MOV     (R0)+,(R5)+     ;PARAMETER BLOCK
 6945  033666  012025                          MOV     (R0)+,(R5)+
 6946  033670  012025                          MOV     (R0)+,(R5)+
 6947  033672  012025                          MOV     (R0)+,(R5)+
 6948  033674  012025                          MOV     (R0)+,(R5)+
 6949  033676  012705  002640                  MOV     #PARM0,R5
 6950  033702  012737  000000  177776          MOV     #PR0,@#PSW      ;LOWER PRIORITY TO ALLOW INTERRUPT
 6951  033710  004737  030124                  JSR     PC,DRVCAL       ;CALL DRIVER
 6952  033714  105037  003135          2$:     CLRB    ERRCNT          ;CLEAR ERROR COUNT
 6953  033720  104410                          RESREG                  ;IF RETURN GETS HERE, NO ERROR
 6954  033722  000207                          RTS     PC              ;OCCURRED, RECOVERY WAS SUCCESSFUL
 6955
 6956
 6957  033724  152737  000377  003132  RETANL: BISB    #377,DONE       ;SET DONE
 6958  033732  052737  000400  005424          BIS     #LEV2ER,RECODE  ;SET LEVEL TWO ERROR
 6959  033740  000207                          RTS     PC
 6960  033742  152737  000377  003132  RETNML: BISB    #377,DONE       ;SET DONE
 6961  033750  000207                          RTS     PC
 6962
 6963
 6964                                  ;;*****************************************************************
 6965                                  .SBTTL  TIME OUT PROCESSOR ROUTINE
 6966                                  ;*THIS ROUTINE SUPPORTS THE ERROR HANDLER BY PROCESSING TIME OUT STATUS
 6967                                  ;*GATHERING DUTIES.
```

```
                                ;;*****************************************************************
033752                          TOPROC:
033752  104407                          SAVREG
033754  013702  003046                  MOV     RKBAS,R2
033760  012701  001174                  MOV     #$REGS,R1       ;SET UP R1 FOR RK REGISTER STORAGE
033764  016221  000000                  MOV     RKCS1(R2),(R1)+ ;STORE ALL VALID RK611
033770  016221  000010                  MOV     RKCS2(R2),(R1)+ ;REGISTERS
033774  016221  000020                  MOV     RKDC(R2),(R1)+
034000  016221  000006                  MOV     RKDA(R2),(R1)+
034004  016221  000002                  MOV     RKWC(R2),(R1)+
034010  016221  000004                  MOV     RKBA(R2),(R1)+
034014  016221  000016                  MOV     RKASOF(R2),(R1)+
034020  016221  000012                  MOV     RKDS(R2),(R1)+
034024  016221  000014                  MOV     RKER(R2),(R1)+
034030  005000                          CLR     R0              ;THIS CODE WILL ATTEMPT TO
                                                                ;RETRIEVE THE STATUS FROM THE
                                                                ;DRIVE.
034032  012705  002724                  MOV     #PARM1,R5       ;SET UP TO USE PARM1
034036  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;DO READ DRIVE STATUS COMMAND
034044  004737  030124                  JSR     PC,DRVCAL       ;CALL DRIVER
034050  032765  000100  000014          BIT     #CMDTO,P.PRST(R5) ;TEST FOR TIMEOUT
034056  001403                          BEQ     1$              ;NO - SKIP
034060  052737  002000  005424          BIS     #TWOTOS,RECODE  ;SET TWO TIMEOUTS FLAG
034066  062705  000040          1$:     ADD     #P.AOO,R5       ;BUMP R5 TO POINT TO DRIVE STATUS
034072  012521                          MOV     (R5)+,(R1)+     ;MOVE ALL THE DRIVE STATUS INTO THE
034074  012521                          MOV     (R5)+,(R1)+     ;TEMP REGS FOR REPORTING.
034076  012521                          MOV     (R5)+,(R1)+
034100  012521                          MOV     (R5)+,(R1)+
034102  012521                          MOV     (R5)+,(R1)+
034104  012521                          MOV     (R5)+,(R1)+
034106  012521                          MOV     (R5)+,(R1)+
034110  012521                          MOV     (R5)+,(R1)+

034112  012705  002640                  MOV     #PARM0,R5       ;RESTORE PARM 0
034116  104410                          RESREG
034120  000207                          RTS     PC


                                ;;*****************************************************************
                                .SBTTL  READ HEADER 0 ROUTINE
                                ;;*****************************************************************
034122                          RDHD0:
034122  104407                          SAVREG
034124  016501  000026                  MOV     P.DTS(R5),R1    ;STORE TRACK AND SECTOR
034130  016500  000052                  MOV     P.B10(R5),R0    ;GET THE CYLINDER ADRS
034134  042700  160017                  BIC     #160017,R0      ;FROM THE DRIVE STATUS. CLEAR
034140  006200                          ASR     R0              ;OFF UNUSED BITS AND POSITION
034142  006200                          ASR     R0              ;FOR USE AS THE DESIRED
034144  006200                          ASR     R0              ;CYLINDER IN THE READ
034146  006200                          ASR     R0              ;HEADER COMMAND.
034150  012705  002724                  MOV     #PARM1,R5       ;SET UP TO USE P.B. 1
034154  010165  000004                  MOV     R1,P.SECT(R5)   ;INSERT TRK,SECT FOR READ HDR
034160  010065  000002                  MOV     R0,P.CYLN(R5)   ;SET CYL NO.
034164  112765  000177  000001          MOVB    #SUBCLR,P.CMND(R5)      ;SET S.S. CLEAR CMND
034172  012737  000000  177776          MOV     #PR0,@#PSW      ;ALLOW INTERRUPTS
```

```
 7024  034200  004737  030124                    JSR     PC,DRVCAL          ;INIT THE S.S.
 7025  034204  112765  000125  000001            MOVB    #RDHEAD,P.CMND(R5) ;SET READ HEADER COMMAND
 7026  034212  004737  030124                    JSR     PC,DRVCAL          ;DO A READ HEADER
 7027  034216  013762  024364  000000            MOV     HOLD1,RKCS1(R2) ;DO A READ HDR, WITH FMT BIT = 0
 7028  034224  032762  000200  000000  24$:      BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
 7029  034232  001774                            BEQ     24$
 7030  034234  013762  024366  000000            MOV     HOLD2,RKCS1(R2) ;DO A READ HDR, WITH FMT BIT = 1
 7031  034242  032762  000200  000000  26$:      BIT     #RDY,RKCS1(R2)  ;WAIT FOR READY
 7032  034250  001774                            BEQ     26$
 7033  034252  142765  000020  000007            BICB    #B.CFMT,P.CS1H(R5) ;CLEAR THE FORMAT BIT
 7034  034260  153765  003134  000007            BISB    FORMAT,P.CS1H(R5) ;RESTORE TYPE AND FMT IN USE
 7035  034266  012705  002640                    MOV     #PARM0,R5          ;RESTORE P.B. 0 ADDRESS
 7036  034272  104410                            RESREG                     ;RESTORE R0-R5
 7037  034274  000207                            RTS     PC                 ;RETURN
 7038
 7039
 7040
 7041                                    ;;*****************************************************************
 7042                                    .SBTTL  GET PACK ADDRESS ROUTINE
 7043                                    ;;*****************************************************************
 7044  034276  016537  000030  001174  GTPKAD: MOV     P.DCYL(R5),$REG5 ;GET CYLINDER NUMBER
 7045  034304  005037  001176                    CLR     $REG6              ;CLEAR REGISTERS FOR
 7046  034310  005037  001200                    CLR     $REG7              ;TRACK & SECTOR STORAGE
 7047  034314  116537  000026  001200            MOVB    P.DTS(R5),$REG7 ;STORE THE TRACK AND SECTOR
 7048  034322  116537  000027  001176            MOVB    P.DTS+1(R5),$REG6
 7049  034330  032737  000004  005424            BIT     #HVRCER,RECODE  ;SEE IF HVRC ERROR
 7050  034336  001034                            BNE     5$              ;BR IF YES
 7051  034340  005737  001200                    TST     $REG7            ;ADJUST THE ADDRESS CONTAINED IN
 7052                                                                     ;THE RK REGISTERS FOR THE AUTOMATIC
 7053  034344  001403                            BEQ     1$              ;INCREMENT
 7054  034346  005337  001200                    DEC     $REG7
 7055  034352  000426                            BR      5$
 7056  034354  032765  010000  000016  1$:       BIT     #CFMT,P.CS1(R5)
 7057  034362  001404                            BEQ     2$
 7058  034364  012737  000023  001200            MOV     #19.,$REG7
 7059  034372  000403                            BR      3$
 7060  034374  012737  000025  001200  2$:       MOV     #21.,$REG7
 7061  034402  005737  001176          3$:       TST     $REG6
 7062  034406  001403                            BEQ     4$
 7063  034410  005337  001176                    DEC     $REG6
 7064  034414  000405                            BR      5$
 7065  034416  012737  000002  001176  4$:       MOV     #2,$REG6
 7066  034424  005337  001174                    DEC     $REG5
 7067  034430  000207                  5$:       RTS     PC
 7068
 7069
 7070
 7071                                    ;;*****************************************************************
 7072                                    .SBTTL  BUILD EXPECTED HEADER
 7073                                    ;*USES DESIRED CYLINDER, TRACK AND SECTOR REGISTERS TO DETERMINE
 7074                                    ;*WHICH HEADER WAS EXPECTED. LOADS EXPECTED VALUES IN $REG5, 6, AND
 7075                                    ;*7 FOR REPORTING.
 7076                                    ;;*****************************************************************
 7077  034432  104407                    BLDEXH: SAVREG
 7078  034434  016537  000030  001174            MOV     P.DCYL(R5),$REG5 ;CONSTRUCT EXPECTED HDR
 7079  034442  016501  000026                    MOV     P.DTS(R5),R1 ;DESIRED CYLINDER & DESIRED TRACK
```

```
 7080  034446  042701  174377                  BIC     #174377,R1      ;CLEAR ALL BUT TRACK BITS
 7081  034452  006201                          ASR     R1              ;AND SECTOR. SHIFT THE TRACK
 7082  034454  0062C1                          ASR     R1              ;OVER TO CONFORM TO HEADER FORMAT
 7083  034456  006201                          ASR     R1              ;CHECK THE FORMAT BIT AND
 7084                                                                  ;IF SET, SET THE HEADER FORMAT
 7085  034460  016537  000026  001176          MOV     P.DTS(R5),$REG6 ;BIT.
 7086  034466  042737  177740  001176          BIC     #177740,$REG6   ;CLEAR ALL BUT SECTOR
 7087  034474  060137  001176                  ADD     R1,$REG6        ;ADD TRACK AND SECTOR TOGETHER
 7088  034500  052737  140000  001176          BIS     #140000,$REG6   ;INSERT BSE BITS
 7089  034506  032765  010000  000016          BIT     #CFMT,P.CS1(R5)
 7090  034514  001403                          BEQ     23$
 7091  034516  052737  001000  001176          BIS     #1000,$REG6
 7092  034524  013737  001174  001200  23$:    MOV     $REG5,$REG7     ;COMPUTE THE HEADER VRC
 7093  034532  013701  001176                  MOV     $REG6,R1
 7094  034536  043737  001176  001200          BIC     $REG6,$REG7
 7095  034544  043701  001174                  BIC     $REG5,R1
 7096  034550  050137  001200                  BIS     R1,$REG7
 7097  034554  104410                          RESREG
 7098  034556  000207                          RTS     PC
 7099
```

```
                                .SBTTL  RK611/RK06-RK07 UNIBUS DRIVER FOR SEQUENTIAL OPERATIONS (REV. 0.10)

                                ;*COPYRIGHT (C) 1975,1976,1977
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MA. 01754
                                ;*AUTHOR: ROY SPITZER

                                .SBTTL   *WATCH-DOG TIMER

                                ;;*****************************************************************
                                ;*
                                ;*      THE WATCH-DOG TIMER DOES A PSEUDO-TIMING OF RK06-RK07 UNIBUS
                                ;*      SUBSYSTEM COMMAND. SINCE ONE CAN NOT GUARANTEE THAT A
                                ;*      REAL-TIME CLOCK (KW11-P OR KW11-L) IS ON THE SYSTEM
                                ;*      THE RK06-RK07 DRIVER WILL USE THE LOCATION W.MTIM FOR
                                ;*      MILLI-SECOND TIMING. WHEN W.MTIM REACHES ZERO THE
                                ;*      WATCH-DOG TIMER WILL SCAN THE DRIVES IN USE AS
                                ;*      DETERMINED BY THE LOCATION W.TIME. THE TIMER COUNTS
                                ;*      (ONE FOR EACH DRIVE) ARE KEPT IN THE TABLE W.DRV.
                                ;*      IF ANY COUNT IN THE TABLE W.DRV REACHES ZERO A COMMAND
                                ;*      TIME-OUT WILL BE DESIGNATED IN THE PROGRAM DEVICE STATUS
                                ;*      REGISTER OF THAT DRIVE'S PARAMETER BLOCK.
                                ;*
                                ;*      THE DRIVER WILL USE THE LOCATION W.MIN AS THE NUMBER
                                ;*      OF MILLISECONDS FOR AN UNLOAD OR START SPINDLE COMMAND.
                                ;*      THE DRIVER WILL USE THE LOCATION W.SEC AS THE TIME
                                ;*      LIMIT FOR ALL OTHER COMMANDS.
                                ;*
                                ;*      FOR QUEUED OPERATIONS THE WATCH-DOG TIMER WILL
                                ;*      WATCH UP TO 8 OPERATIONS SIMULATEOUSLY. FOR SEQUENTIAL
                                ;*      OPERATIONS ONLY ONE OPERATION WILL BE WATCHED.
                                ;*
                                ;*CALL  JSR     PC,W.WTCH
                                ;*      RETURN          IF NO DRIVE ORDER EXCEEDED ITS TIME LIMIT
                                ;*
                                ;*      OTHERWISE AN ABNORMAL RETURN TO THE ROUTINE ADDRESS
                                ;*      BY LOCATION A.ABNL WILL OCCUR AND THE CMDTO FLAG
                                ;*      IN THE PROGRAM DEVICE STATUS REGISTER OF THE
                                ;*      APPROPRIATE PARAMETER BLOCK WILL BE SET.
                                ;*
                                ;;*****************************************************************

 034560  010546                W.WTCH: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
 034562  010446                        MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
 034564  010346                        MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
 034566  010246                        MOV     R2,-(SP)        ;SAVE R2 ON STACK
 034570  013746  177776                MOV     PS,-(SP)        ;SAVE PROGRAM STATUS WORD ON STACK
 034574  005337  003066                DEC     W.MTIM          ;DECREMENT MILLISECOND TIMER
 034600  001034                        BNE     20$             ;IF NOT ZERO RETURN
 034602  013737  003070  003066        MOV     W.MILI,W.MTIM   ;REINITIALIZE MILLISECOND TIMER
 034610  105737  003110                TSTB    W.TIME          ;CHECK IF DRIVE IS BEING TIMED
 034614  001426                        BEQ     20$             ;NO, RETURN
 034616  013737  003052  177776        MOV     RKPRI,PS        ;LOCK OUT RK06-RK07 INTERRUPTS
 034624  013702  003046                MOV     RKBAS,R2        ;LOAD BASE OF RK06-RK07 REGISTERS
 034630  005337  003124                DEC     W.DRV           ;DECREMENT COMMAND TIMER
 034634  001016                        BNE     20$             ;RETURN IF NO TIME OUT
```

```
  7156  034636  105037  003110                  CLRB    W.TIME          ;RESET TIMING INDICATOR
  7157  034642  013705  003122                  MOV     PBLKT,R5        ;LOAD ADDRESS OF PARAMETER BLOCK
  7158                                                                  ;  TABLE FOR INDEXING
  7159  034646  052765  000100  000014          BIS     #CMDTO,P.PRST(R5) ;SET COMMAND TIME OUT
  7160  034654  020537  003064                  CMP     R5,O.WAIT       ;CHECK IF DRIVER IS WAITING FOR
  7161                                                                  ;  COMMAND COMPLETION
  7162  034660  001002                          BNE     5$              ;NO, DO NOT ALTER WAITING FOR
  7163                                                                  ;  COMMAND COMPLETION
  7164  034662  005037  003064                  CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
  7165  034666  004737  040142          5$:     JSR     PC,R.ABNL       ;BRANCH TO ERROR ROUTINE
  7166  034672  012637  177776          20$:    MOV     (SP)+,PS        ;RESTORE PSW
  7167  034676  012602                          MOV     (SP)+,R2        ;RESTORE R2
  7168  034700  012603                          MOV     (SP)+,R3        ;RESTORE R3
  7169  034702  012604                          MOV     (SP)+,R4        ;RESTORE R4
  7170  034704  012605                          MOV     (SP)+,R5        ;RESTORE R5
  7171  034706  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL   *RK06 INTERRUPT SERVICE ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL SERVICE ALL RK06 INTERRUPTS.
                                ;*
                                ;*      UPON RECEIVING AN INTERRUPT, THIS ROUTINE WILL
                                ;*      PERFORM ONE OF THE FOLLOWING SERVICES:
                                ;*
                                ;*      1.) SERVICE PORT WAS SEIZED BY OTHER PORT
                                ;*      2.) SERVICE DRIVER IS WAIT FOR COMMAND COMPLETION
                                ;*      3.) SERVICE POSITIONING COMPLETION
                                ;*      4.) REQUEUE COMMAND IF DRIVE WAS RELEASED
                                ;*          FOR THE QUEUED RK06 DRIVER.
                                ;*      5.) IF NO SERVICE IS REQUIRED, THE COMMAND WILL BE ISSUED
                                ;*          FOR THE QUEUED RK06 DRIVER.
                                ;*
                                ;*      THREE LINKS ARE PROVIDED TO THE DRIVING PROGRAM.
                                ;*      THEY ARE:
                                ;*
                                ;*      1.) A.NORM ADDRESS OF NORMAL RETURN (SUCESSFUL COMPLETION OF COMMAND)
                                ;*      2.) A.ABNL ADDRESS OF ABNORMAL RETURN (UNSUCESSFUL COMPLETION OF COMMAND)
                                ;*      3.) A.CONT ADDRESS OF CONTROL ERROR RETURN
                                ;*
                                ;*      FOR NORMAL AND ABNORMAL RETURNS, THE ADDRESS OF THE APPROPRIATE
                                ;*      PARAMETER BLOCK WILL BE IN R5.
                                ;*
                                ;*      FOR THE CONTROLLER ERROR RETURN, THE LOCATION E.CONT CONTAINS
                                ;*      THE REASON FOR THE CONTROLLER ERROR.
                                ;*
                                ;*      ROUTINES USED:
                                ;*              C.OPT   (QUEUED ONLY)
                                ;*              Q.PUSH (QUEUED ONLY)
                                ;*              Q.RMOV (QUEUED ONLY)
                                ;*              R.CONT (SEQUENTIAL ONLY)
                                ;*              R.NORM (SEQUENTIAL ONLY)
                                ;*              R.ABNL (SEQUENTIAL ONLY)
                                ;*              I.CSTS
                                ;*              I.STAT
                                ;*              I.ISSU
                                ;*              I.CCLR
                                ;*
                                ;;*****************************************************************

 034710  010546          I.INTR: MOV     R5,-(SP)          ;STORE R5 ON THE STACK
 034712  010446                  MOV     R4,-(SP)          ;STORE R4 ON THE STACK
 034714  010346                  MOV     R3,-(SP)          ;STORE R3 ON THE STACK
 034716  010246                  MOV     R2,-(SP)          ;STORE R2 ON THE STACK
 034720  010146                  MOV     R1,-(SP)          ;STORE R1 ON THE STACK
 034722  010046                  MOV     R0,-(SP)          ;STORE R0 ON THE STACK
 034724  013702  003046          MOV     RKBAS,R2          ;LOAD R2 TO ADDRESS RK06 REGISTER
 034730  016237  000010  003012  MOV     RKCS2(R2),T.CS2   ;STORE CS2
 034736  032737  001000  003012  BIT     #MDS,T.CS2        ;CHECK IF MULTIPLE DRIVE SELECT
 034744  001407                  BEQ     1$                ;NO, CONTINUE PROCESSING
 034746  052737  100000  003062  BIS     #E.MDS,E.CONT     ;SET MULTIPLE DRIVE SELECT
 034754  004737  040166          JSR     PC,R.CONT         ;REPORT ERROR
```

```
 7228  034760  000137  037140                    JMP     I.RTRN          ;RETURN
 7229
 7230  034764  105737  003104          1$:       TSTB    I.ISRL          ;CHECK IF INTERRUPT OR RELEASE
 7231  034770  001410                             BEQ     6$              ;NO, CHECK IF DRIVE AVAILABLE
 7232  034772  100403                             BMI     5$              ;CHECK IF RELEASE COMMAND
 7233  034774  105037  003104                     CLRB    I.ISRL          ;YES,CLEAR FLAG
 7234  035000  000473                             BR      I.100           ;CONTINUE PROCESSING INTERRUPT
 7235
 7236  035002  105037  003104          5$:       CLRB    I.ISRL          ;CLEAR FLAG
 7237  035006  000137  036122                     JMP     I.ATTN          ;GO PROCESS DRIVE ATTENTIONS
 7238
 7239  035012  032737  010400  003012  6$:       BIT     #NED!UFE,T.CS2  ;CHECK FOR NON-EXISTENT DRIVE OR
 7240                                                                     ;  UNIT FIELD ERROR
 7241  035020  001413                             BEQ     7$              ;NO, WAIT FOR DUAL ACCESS INTERRUPT
 7242  035022  013704  003012                     MOV     T.CS2,R4        ;LOAD R4 FOR DRIVE NUMBER
 7243  035026  042704  177770                     BIC     #^C<DRVMSK>,R4  ;KEEP DRIVE BITS
 7244  035032  013705  003122                     MOV     PBLKT,R5        ;STORE PARAMETER BLOCK ADDRESS
 7245  035036  016237  000000  003010             MOV     RKCS1(R2),T.CS1 ;LOAD TEMPORARY CS1 FOR STATUS REPORT
 7246  035044  000137  035332                     JMP     I.ERRC          ;REPORT ERROR
 7247
 7248  035050  016237  000012  003030  7$:       MOV     RKDS(R2),T.DS   ;STORE STATUS REGISTER FOR COMPARISON
 7249  035056  032737  000001  003030             BIT     #DRA,T.DS       ;CHECK IF DRIVE SEIZED BY OTHER
 7250                                                                     ;  PORT
 7251  035064  001041                             BNE     I.100           ;NO, CONTINUE PROCESSING INTERRUPT
 7252
 7253                                             ;CHECK IF ANY DATA TRANSFER ERROR EXISTS
 7254  035066  032737  164000  003012             BIT     #DLT!WCE!UPE!NEM,T.CS2
 7255
 7256  035074  001007                             BNE     10$             ;INDICATE ERROR
 7257  035076  016237  000014  003026             MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
 7258
 7259                                     ;       CHECK FOR DATA TRANSFER ERROR TYPE ERROR
 7260  035104  032737  125700  003026             BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
 7261
 7262  035112  001407                             BEQ     11$             ;NO, WAIT FOR RELEASE OF RK06 DRIVE
 7263
 7264  035114  052737  000010  003062  10$:       BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
 7265  035122  004737  040166                     JSR     PC,R.CONT       ;REPORT ERROR
 7266  035126  000137  037140                     JMP     I.RTRN          ;RESTORE REGISTERS
 7267
 7268  035132  105037  003110          11$:      CLRB    W.TIME          ;RESET TIMING ON THIS DRIVE
 7269  035136  005037  003124                     CLR     W.DRV           ;CLEAR TIMING COUNT FOR THIS DRIVE
 7270  035142  013705  003122                     MOV     PBLKT,R5        ;LOAD R5 WITH PARAMETER BLOCK
 7271                                                                     ;  ADDRESS
 7272  035146  052765  010000  000014             BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SEIZED IN THE
 7273                                                                     ;  PROGRAM DRIVE STATUS REGISTER
 7274  035154  005037  003064                     CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
 7275  035160  004737  040142                     JSR     PC,R.ABNL       ;INDICATE ABNORMAL TERMINATION
 7276  035164  000137  037140                     JMP     I.RTRN          ;GO RESTORE REGISTERS
 7277
 7278  035170  013705  003064          I.100:    MOV     O.WAIT,R5       ;LOAD PARAMETER BLOCK ADDRESS INTO R5
 7279  035174  001002                             BNE     2$              ;IS COMMAND WAITING PROCESSING
 7280                                                                     ;  YES, DO PROCESSING
 7281  035176  000137  036122                     JMP     I.ATTN          ;NO, PROCESS ATTENTION
 7282
 7283  035202  013704  003012          2$:       MOV     T.CS2,R4        ;STORE RKCS2 FOR DRIVE NUMBER
```

```
 7284  035206  042704  177770                    BIC     #^C<DRVMSK>,R4  ;MASK OUT UNNECESSARY BITS
 7285
 7286
 7287  035212  126504  000000                    CMPB    P.DRVN(R5),R4   ;CHECK IF DRIVE NUMBER IS EXPECTED
 7288  035216  001401                            BEQ     3$              ;YES, CONTINUE
 7289  035220  000000                            HALT                    ;NO, DRIVER ERROR
 7290  035222  122765  000164  000001  3$:       CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 7291  035230  001002                            BNE     10$             ;NO, EXECUTE NORMAL DATA TRANSFER
 7292  035232  000137  035570                    JMP     I.HDAL          ;GO EXECUTE SPECIAL HEADER SEQUENCE
 7293
 7294  035236  005037  003064          10$:     CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
 7295  035242  005037  003124                    CLR     W.DRV           ;CLEAR WATCH-DOG TIME
 7296  035246  105037  003110                    CLRB    W.TIME          ;RESET TIMING ON THIS DRIVE
 7297  035252  016237  000000  003010            MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
 7298  035260  032737  100000  003010            BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 7299  035266  001021                            BNE     I.ERRC          ;YES, PROCESS ERROR
 7300  035270  016237  000016  003024            MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 7301  035276  133737  003111  003025            BITB    INTMSK,T.ASOF+1 ;CHECK IF DRIVE ATTENTION SET
 7302  035304  001004                            BNE     15$             ;YES, REPORT ERROR
 7303  035306  004737  040154                    JSR     PC,R.NORM       ;INDICATE NORMAL RETURN
 7304  035312  000137  037140                    JMP     I.RTRN          ;RESTORE REGISTERS
 7305
 7306  035316  052765  000010  000014  15$:     BIS     #UEXATT,P.PRST(R5) ;SET UNEXPECTED ATTEN ION
 7307
 7308  035324  004737  037610          I.ERRA:  JSR     PC,I.CSTS       ;STORE CONTROLLER STATUS
 7309  035330  000405                            BR      I.ERR           ;STORE PATTERN AND POSITION INFORMATION
 7310
 7311  035332  013765  003010  000016  I.ERRC:  MOV     T.CS1,P.CS1(R5) ;GET ERROR RKCS1
 7312  035340  004737  037632                    JSR     PC,I.CST1       ;GET REST OF CONTROLLER STATUS
 7313  035344  016265  000032  000062  I.ERR:   MOV     RKECPT(R2),P.EPAT(R5) ;STORE ECC PATTERN
 7314  035352  016265  000030  000060            MOV     RKECPS(R2),P.EPOS(R5) ;STORE ECC POSITION
 7315  035360  004037  037156                    JSR     R0,I.CCLR       ;CLEAR CONTROLLER
 7316  035364  037140                            I.RTRN                  ;ERROR RETURN
 7317  035366  032765  010400  000020            BIT     #NED!UFE,P.CS2(R5) ;CHECK IF IT WAS NON-EXISTENT DRIVE OR
 7318                                                                    ;  UNIT FIELD ERROR
 7319  035374  001046                            BNE     5$              ;YES, REPORT ERROR
 7320  035376  004037  037714                    JSR     R0,I.STAT       ;GATHER DRIVE STATUS
 7321  035402  037140                            I.RTRN                  ;ERROR RETURN
 7322  035404  112737  000005  003010            MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
 7323  035412  004037  037240                    JSR     R0,I.ISSU       ;ISSUE DRIVE CLEAR
 7324  035416  037140                            I.RTRN                  ;ERROR RETURN
 7325  035420  133737  003111  003025            BITB    INTMSK,T.ASOF+1 ;CHECK IF ATTENTION RESET
 7326  035426  001407                            BEQ     2$              ;NO, INDICATE DRIVE ERROR
 7327  035430  052737  000020  003062            BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
 7328                                                                    ;  WITH CLEAR
 7329  035436  004737  040166                    JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
 7330  035442  000137  037140                    JMP     I.RTRN          ;GO RESTORE REGISTERS
 7331
 7332  035446  032737  040000  003034  2$:      BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE CLEARED
 7333  035454  001403                            BEQ     3$              ;YES, CHECK FAULT
 7334  035456  052765  000040  000014            BIS     #DRVDSC,P.PRST(R5) ;SET DSC DID NOT CLEAR
 7335  035464  032737  001000  003036  3$:      BIT     #S.PAR,T.MR3    ;CHECK IF DRIVE PARITY ERROR
 7336  035472  001407                            BEQ     5$              ;NO, INDICATE ABNORMAL TERMINATION
 7337  035474  052737  002000  003062            BIS     #E.DPAR,E.CONT  ;SET DRIVE PARITY ERROR
 7338  035502  004737  040166                    JSR     PC,R.CONT       ;INDICATE CONTROLLER ERROR
 7339  035506  000137  037140                    JMP     I.RTRN          ;RETURN
```

```
 7340
 7341  035512  032765  000020  000014  5$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF HARD DRIVE ERROR
 7342  035520  001017                          BNE     10$                ;YES, GO REPORT ERROR
 7343  035522  032737  020000  003034          BIT     #S.PIP,T.MR2       ;CHECK IF DRIVE IS CYCLING DOWN
 7344  035530  001413                          BEQ     10$                ;NO, REPORT ERROR
 7345  035532  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING
 7346  035540  113737  003111  003110          MOVB    INTMSK,W.TIME      ;SET UP 8 SECONDS FOR DRIVE TO CYCLE UP
 7347  035546  013737  003074  003124          MOV     W.8SEC,W.DRV
 7348  035554  000137  037140                  JMP     I.RTRN             ;GO RESTORE REGISTERS
 7349
 7350  035560  004737  040142          10$:    JSR     PC,R.ABNL          ;GO REPORT ERROR
 7351  035564  000137  037140                  JMP     I.RTRN             ;GO RESTORE REGISTERS
 7352
 7353                                  .SBTTL   *READ ALL HEADERS INTERRUPT SEQUENCE
 7354
 7355  035570  016237  000000  003010  I.HDAL: MOV     RKCS1(R2),T.CS1 ;STORE CS1 TO CHECK CONTROLLER
 7356                                                                  ;  ERROR
 7357  035576  032737  100000  003010          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 7358  035604  001422                          BEQ     5$              ;NO, CHECK FOR ATTENTION
 7359
 7360  035606  005037  003064                  CLR     O.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETE
 7361  035612  105037  003110                  CLRB    W.TIME          ;RESET TIMING ON DRIVE
 7362  035616  005037  003124                  CLR     W.DRV           ;CLEAR TIME OUT COUNT
 7363  035622  013765  003010  000016          MOV     T.CS1,P.CS1(R5) ;STORE ERROR RKCS1
 7364  035630  004737  037632                  JSR     PC,I.ST1        ;STORE CONTROLLER REGISTERS
 7365  035634  004037  037156                  JSR     R0,I.CCLR       ;CLEAR CONTROLLER
 7366  035640  037140                          I.RTRN                  ;ERROR RETURN
 7367  035642  004737  040142                  JSR     PC,R.ABNL       ;INDICATE ERROR RETURN
 7368  035646  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
 7369
 7370  035652  016237  000016  003024  5$:     MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 7371  035660  133737  003111  003025          BITB    INTMSK,T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS SET
 7372  035666  001410                          BEQ     7$              ;NO, CHECK IF READ ALL HEADERS
 7373  035670  005037  003064                  CLR     O.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETION
 7374  035674  105037  003110                  CLRB    W.TIME          ;RESET TIMING ON DRIVE
 7375  035700  005037  003124                  CLR     W.DRV           ;CLEAR TIME OUT COUNT
 7376  035704  000137  035324                  JMP     I.ERRA          ;GO REPORT ERROR
 7377
 7378  035710  013701  003100          7$:     MOV     HDR.AD,R1       ;GET MAIN MEMORY ADDRESS
 7379  035714  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET FIRST WORD OF HEADER
 7380  035720  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET SECOND WORD OF HEADER
 7381  035724  016221  000024                  MOV     RKDB(R2),(R1)+  ;GET THIRD WORD OF HEADER
 7382  035730  010137  003100                  MOV     R1,HDR.AD       ;STORE ADDRESS FOR NEXT HEADER
 7383  035734  016237  000010  003012          MOV     RKCS2(R2),T.CS2 ;STORE CS2 TO CHECK FOR DATA LATE
 7384  035742  032737  100000  003012          BIT     #DLT,T.CS2      ;CHECK FOR DATA LATE
 7385  035750  001055                          BNE     35$             ;YES, REPORT ERROR
 7386  035752  005337  003102                  DEC     HDR.CT          ;DECREMENT NUMBER OF HEADER YET TO READ
 7387  035756  001026                          BNE     25$             ;IF NON-ZERO, GO ISSUE NEXT READ HEADER
 7388  035760  005037  003064                  CLR     O.WAIT          ;CLEAR DRIVER WAITING FOR COMMAND COMPLETION
 7389  035764  005037  003124                  CLR     W.DRV           ;CLEAR TIME OUT COUNT FOR THIS DRIVE
 7390  035770  105037  003110                  CLRB    W.TIME          ;CLEAR WATCH DOG TIME ON THIS DRIVE
 7391  035774  012762  000003  000026          MOV     #3,RKMR1(R2)    ;LOAD MAINTENANCE REGISTER FOR SECTOR COUNT
 7392  036002  112737  000001  003010          MOVB    #DR.SEL,T.CS1   ;LOAD SELECT COMMAND
 7393  036010  004037  037240                  JSR     R0,I.ISSU       ;GET SECTOR COUNT
 7394  036014  037140                          I.RTRN                  ;ERROR RETURN
 7395  036016  013765  003036  000056          MOV     T.MR3,P.B11(R5) ;LOAD SECTOR COUNT
```

```
7396  036024  004737  040154                  JSR     PC,R.NORM       ;INDICATE NORMAL TERMINATION
7397  036030  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
7398
7399  036034  016562  000002  000020  25$:    MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
7400  036042  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
7401  036050  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
7402  036056  042765  165777  000016          BIC     #^C<CDT!CFMT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT AND
7403                                                                  ;  DRIVE TYPE
7404  036064  112765  000125  000016          MOVB    #RDHEAD,P.CS1(R5)  ;STORE COMMAND ISSUED
7405  036072  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2)  ;ISSUE READ HEADER
7406  036100  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
7407
7408  036104  052737  000400  003062  35$:    BIS     #E.DLT,E.CONT   ;SET DATA LATE WHILE UNLOADING HEADER
7409  036112  004737  040166                  JSR     PC,R.CONT       ;REPORT ERROR
7410  036116  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
7411
7412                                  .SBTTL   *DRIVE ATTENTION SCANNER
7413
7414  036122  016237  000000  003010  I.ATTN: MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS
7415                                                                  ;  REGISTER 1 FOR COMPARISON
7416  036130  032737  100000  003010          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCURRED
7417  036136  001441                          BEQ     5$              ;NO, CHECK IF ATTENTION
7418
7419                                          ;CHECK IF ANY DATA TRANSFER TYPE ERROR EXISTS
7420  036140  032737  164000  003012          BIT     #DLT!WCE!UPE!NEM,T.CS2
7421
7422  036146  001007                          BNE     1$              ;INDICATE ERROR
7423  036150  016237  000014  003026          MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
7424
7425                                  ;       CHECK FOR DATA TRANSFER ERROR TYPE
7426  036156  032737  125700  003026          BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
7427
7428  036164  001407                          BEQ     2$              ;NO DATA TRANSFER ERROR
7429
7430  036166  052737  000010  003062  1$:     BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
7431  036174  004737  040166                  JSR     PC,R.CONT       ;REPORT ERROR
7432  036200  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
7433
7434  036204  013704  003012          2$:     MOV     T.CS2,R4        ;SAVE CS2 FOR REGISTER NUMBER
7435  036210  042704  177770                  BIC     #^C<DRVMSK>,R4  ;STRIP OFF JUNK
7436  036214  105037  003110                  CLRB    W.TIME          ;CLEAR WATCH DOG TIMER
7437  036220  005037  003124                  CLR     W.DRV           ;RESET TIMER VALUE
7438  036224  013705  003122                  MOV     PBLKT,R5        ;STORE PARAMETER BLOCK ADDRESS IN R5
7439
7440                                  ;       CLEAR DRIVE POSITIONING AND DRIVE POSITIONED FOR DATA TRANSFER
7441                                  ;         IN PROGRAM DEVICE STATUS REGISTER
7442  036230  042765  000006  000014          BIC     #DRVPOS!DRVPDT,P.PRST(R5)
7443
7444  036236  000137  035332                  JMP     I.ERRC          ;GO REPORT ERROR
7445
7446  036242  032737  040000  003010  5$:     BIT     #DI,T.CS1       ;CHECK IF ANY DRIVE ATTENTION
7447  036250  001002                          BNE     6$              ;YES, PROCESS INTERRUPT
7448  036252  000137  037140                  JMP     I.RTRN          ;RESTORE REGISTERS
7449
7450  036256  016237  000016  003024  6$:     MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
7451  036264  105737  003025                  TSTB    T.ASOF+1        ;CHECK IF ANY ATTENTIONS SET
```

```
 7452  036270  001007                          BNE     7$                ;YES GO PROCESS INTERRUPT
 7453  036272  052737  000002  003062          BIS     #E.NOAT,E.CONT    ;SET NO ATTENTION IN ATTENTION SUMMARY
 7454  036300  004737  040166                  JSR     PC,R.CONT         ;GO REPORT ERROR
 7455  036304  000137  037140                  JMP     I.RTRN            ;GO RESTORE REGISTERS
 7456
 7457  036310  133737  003111  003025  7$:     BITB    INTMSK,T.ASOF+1   ;CHECK IF DESIRED INTERRUPT
 7458  036316  001007                          BNE     8$                ;YES, GO PROCESS IT
 7459  036320  052737  000004  003062          BIS     #E.UATT,E.CONT    ;SET UNSOLICATED ATTENTION
 7460  036326  004737  040166                  JSR     PC,R.CONT         ;GO REPORT ERROR
 7461  036332  000137  037140                  JMP     I.RTRN            ;GO RESTORE REGISTERS
 7462
 7463  036336  013705  003122          8$:     MOV     PBLKT,R5          ;STORE PARAMETER BLOCK TABLE
 7464  036342  116504  000000                  MOVB    P.DRVN(R5),R4     ;STORE DRIVE NUMBER
 7465  036346  032765  020000  000014          BIT     #E.UNLD,P.PRST(R5)  ;CHECK IF DRIVE UNLOADING
 7466  036354  001402                          BEQ     11$               ;NO, CONTINUE
 7467  036356  000137  037040                  JMP     I.UNLD            ;SERVICE DRIVE IN POSITION AFTER ERROR
 7468
 7469  036362  042765  000002  000014  11$:     BIC     #DRVPOS,P.PRST(R5)  ;RESET DRIVE POSITIONING
 7470  036370  005062  000026                  CLR     RKMR1(R2)         ;CLEAR MAINTENANCE REGISTER 1
 7471  036374  112737  000001  003010          MOVB    #DR.SEL,T.CS1     ;LOAD COMMAND
 7472  036402  004037  037240                  JSR     R0,I.ISSU         ;SELECT DRIVE WITH ATTENTION HIGH
 7473  036406  037140                          I.RTRN                    ;ERROR RETURN
 7474  036410  013765  003036  000042          MOV     T.MR3,P.B00(R5)   ;STORE STATUS BYTE 00 MESS B
 7475  036416  032765  000200  000042          BIT     #S.FLT,P.B00(R5)  ;CHECK IF DRIVE FAULT
 7476  036424  001401                          BEQ     12$               ;NO, CHECK FOR DRIVE STATUS CHANGE
 7477  036426  000461                          BR      I.AERR            ;PROCESS ERROR
 7478
 7479  036430  013765  003034  000040  12$:     MOV     T.MR2,P.A00(R5)   ;STORE MAINTENANCE REGISTER 2
 7480  036436  032765  040000  000040          BIT     #S.DSC,P.A00(R5)  ;CHECK FOR DRIVE STATUS CHANGE
 7481  036444  001004                          BNE     13$               ;YES, PROCESS DRIVE STATUS CHANGE
 7482  036446  052765  004000  000014          BIS     #NODSC,P.PRST(R5) ;SET NO DRIVE STATUS CHANGE
 7483  036454  000446                          BR      I.AERR            ;PROCESS ERROR
 7484
 7485  036456  112737  000005  003010  13$:     MOVB    #DR.CLR,T.CS1     ;LOAD COMMAND
 7486  036464  004037  037240                  JSR     R0,I.ISSU         ;CLEAR DRIVE STATUS CHANGE
 7487  036470  037140                          I.RTRN                    ;ERROR RETUN
 7488  036472  013765  003024  000032          MOV     T.ASOF,P.ASOF(R5) ;STORE ATTENTION SUMMARY
 7489  036500  133765  003111  000033          BITB    INTMSK,P.ASOF+1(R5) ;CHECK IF ATTENTION RESET
 7490  036506  001407                          BEQ     15$               ;YES, CONTINUE INTERRUPT PROCESSING
 7491  036510  052737  000020  003062          BIS     #E.CLAT,E.CONT    ;SET ATTENTION DID NOT RESET
 7492                                                                    ;  WITH DRIVE CLEAR
 7493  036516  004737  040166                  JSR     PC,R.CONT         ;FLAG ERROR
 7494  036522  000137  037140                  JMP     I.RTRN            ;RESTORE REGISTERS
 7495
 7496  036526  013765  003034  000040  15$:     MOV     T.MR2,P.A00(R5)   ;STORE MAINTENANCE REGISTER 2
 7497  036534  032765  040000  000040          BIT     #S.DSC,P.A00(R5)  ;CHECK IF DRIVE STATUS CHANGE
 7498                                                                    ;  RESET
 7499  036542  001404                          BEQ     16$               ;YES, CONTINUE INTERRUPT PROCESSING
 7500  036544  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STATUS CHANGE DID NOT CLEAR
 7501  036552  000407                          BR      I.AERR            ;GO PROCESS ERROR
 7502
 7503  036554  105037  003110          16$:     CLRB    W.TIME            ;RESET TIMING ON THIS DRIVE
 7504  036560  005037  003124                  CLR     W.DRV             ;CLEAR DRIVE TIMING COUNT
 7505  036564  004737  040154                  JSR     PC,R.NORM         ;REPORT SUCCESSFUL COMMAND COMPLETETION
 7506  036570  000563                          BR      I.RTRN            ;RESTORE REGISTERS
 7507
```

```
 7508                                           .SBTTL   *ATTENTION ERROR HANDLER
 7509
 7510  036572  042765  000004  000014  I.AERR: BIC     #DRVPDT,P.PRST(R5) ;RESET POSITIONING IN PROGRESS BECAUSE
 7511                                                                     ;  OF DATA TRANSFER
 7512  036600  105037  003110                  CLRB    W.TIME             ;CLEAR TIMING FOR THIS DRIVE
 7513  036604  005037  003124                  CLR     W.DRV              ;RESET WATCH-DOG TIME
 7514  036610  042765  177741  000016          BIC     #177741,P.CS1(R5)  ;KEEP COMMAND ISSUED
 7515  036616  042737  000036  003010          BIC     #36,T.CS1          ;KEEP CURRENT CONTROLLER STATUS
 7516  036624  053765  003010  000016          BIS     T.CS1,P.CS1(R5)    ;MAKE GOOD MESSAGE
 7517  036632  013765  003012  000020          MOV     T.CS2,P.CS2(R5)    ;STORE CONTROLLER REGISTERS
 7518  036640  013765  003014  000022          MOV     T.WCR,P.WCR(R5)
 7519  036646  013765  003016  000024          MOV     T.BA,P.BAR(R5)
 7520  036654  013765  003020  000026          MOV     T.DA,P.DTS(R5)
 7521  036662  013765  003022  000030          MOV     T.DC,P.DCYL(R5)
 7522  036670  013765  003024  000032          MOV     T.ASOF,P.ASOF(R5)
 7523  036676  013765  003026  000034          MOV     T.ER,P.ER(R5)
 7524  036704  013765  003030  000036          MOV     T.DS,P.DS(R5)
 7525  036712  004037  037714                  JSR     R0,I.STAT          ;GATHER DRIVE STATUS
 7526  036716  037140                          I.RTRN                     ;ERROR RETURN
 7527  036720  112737  000005  003010          MOVB    #DR.CLR,T.CS1      ;LOAD COMMAND
 7528  036726  004037  037240                  JSR     R0,I.ISSU          ;CLEAR DRIVE ERRORS
 7529  036732  037140                          I.RTRN                     ;ERROR RETURN
 7530  036734  133737  003111  003025          BITB    INTMSK,T.ASOF+1    ;CHECK IF ATTENTION RESET
 7531  036742  001407                          BEQ     2$                 ;YES, FLAG DRIVE ERROR
 7532  036744  052737  000020  003062          BIS     #E.CLAT,E.CONT     ;SET ATTENTION DID NOT RESET
 7533  036752  004737  040166                  JSR     PC,R.CONT          ;REPORT ERROR
 7534  036756  000137  037140                  JMP     I.RTRN             ;RESTORE REGISTERS
 7535
 7536  036762  032765  000020  000014  2$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF AHARD DRIVE ERROR
 7537  036770  001017                          BNE     10$                ;YES, REPORT ERROR
 7538  036772  032737  020000  003034          BIT     #S.PIP,T.MR2       ;CHECK IF DRIVE IS UNLOADING
 7539  037000  001413                          BEQ     10$                ;NO, REPORT ERROR
 7540  037002  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING DUE TO ERROR
 7541  037010  113737  003111  003110          MOVB    INTMSK,W.TIME      ;SET TIMING ON THIS DRIVE
 7542  037016  013737  003074  003124          MOV     W.8SEC,W.DRV       ;LOAD 8 SECONDS FOR CYCLE UP TIME
 7543  037024  000137  037140                  JMP     I.RTRN             ;RESTORE REGISTERS
 7544
 7545  037030  004737  040142          10$:    JSR     PC,R.ABNL          ;REPORT ERROR
 7546  037034  000137  037140                  JMP     I.RTRN             ;RESTORE REGISTERS
 7547
 7548                                           .SBTTL   *ERROR CAUSING DRIVE TO UNLOAD
 7549
 7550  037040  052765  020000  000014  I.UNLD: BIS     #E.UNLD,P.PRST(R5) ;CLEAR DRIVE UNLOADING BECAUSE OF ERROR
 7551  037046  112737  000005  003010          MOVB    #DR.CLR,T.CS1      ;LOAD IN DRIVE CLEAR
 7552  037054  004037  037240                  JSR     R0,I.ISSU          ;GO ISSUE DRIVE CLEAR
 7553  037060  037140                          I.RTRN                     ;ERROR RETURN
 7554  037062  136437  003111  003025          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
 7555  037070  001406                          BEQ     15$                ;YES, CONTINUE
 7556  037072  012737  000020  003062          MOV     #E.CLAT,E.CONT     ;SET ATTENTION DID NOT RESET
 7557  037100  004737  040166                  JSR     PC,R.CONT          ;REPORT ERROR
 7558  037104  000415                          BR      I.RTRN             ;RESTORE REGISTERS
 7559
 7560  037106  032737  040000  003034  15$:    BIT     #S.DSC,T.MR2       ;CHECK IF DRIVE STAUS CHANGE RESET
 7561  037114  001403                          BEQ     20$                ;YES, CONTINUE
 7562  037116  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STAUS CHANGE DID NOT CLEAR
 7563  037124  105037  003110          20$:    CLRB    W.TIME             ;RESET TIMING ON THIS DRIVE
```

```
  7564  037130  005037  003124                    CLR     W.DRV           ;CLEAR TIME COUNT
  7565  037134  004737  040142                    JSR     PC,R.ABNL       ;REPORT ERROR
  7566
  7567  037140  012600                  I.RTRN:   MOV     (SP)+,R0        ;RESTORE R0
  7568  037142  012601                            MOV     (SP)+,R1        ;RESTORE R1
  7569  037144  012602                            MOV     (SP)+,R2        ;RESTORE R2
  7570  037146  012603                            MOV     (SP)+,R3        ;RESTORE R3
  7571  037150  012604                            MOV     (SP)+,R4        ;RESTORE R4
  7572  037152  012605                            MOV     (SP)+,R5        ;RESTORE R5
  7573  037154  000002                            RTI                     ;RETURN
  7574
```

```
                                        .SBTTL   *CONTROLLER CLEAR ROUTINE

                                        ;;*****************************************************************
                                        ;*
                                        ;*      THIS ROUTINE WILL BE USED BY THE DRIVER TO CLEAR THE CONTROLLER
                                        ;*      AND CHECK IF THE CONTROLLER ERRORS ARE RESET.  IF THE ERROR IS NOT
                                        ;*      CLEARED, THE ROUTINE AS SPECIFIED IN A.CONT WILL BE CALLED WITH
                                        ;*      E.CCLR SET IN E.CONT.
                                        ;*
                                        ;*      REGISTER        USE
                                        ;*      --------        ---
                                        ;*
                                        ;*      R2              ADDRESS OF RK06 REGISTERS
                                        ;*      R5              ADDRESS OF PARAMETER BLOCK
                                        ;*
                                        ;*CALL  JSR     R0,I.CCLR
                                        ;*      <ADDRESS OF ERROR RETURN>
                                        ;*      RETURN
                                        ;*
                                        ;;*****************************************************************

037156  012762  100000  000000  I.CCLR: MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
037164  016237  000000  003010          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
037172  032737  100000  003010          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER CLEAR DID
                                                                ;  CLEAR ERROR
037200  001407                          BEQ     5$              ;YES, RETURN TO DRIVER PROCESSING
037202  052737  000001  003062          BIS     #E.CCLR,E.CONT  ;SET CLEAR CONTROLLER DID NOT CLEAR ERROR
037210  004737  040166                  JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
037214  011000                          MOV     (R0),R0         ;SET UP ERROR RETURN
037216  000200                          RTS     R0              ;RETURN

037220  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
037226  112737  177777  003104          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE ISSUED
037234  005720                          TST     (R0)+           ;ADJUST FOR NORMAL RETURN
037236  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL   *COMMAND ISSUED BY DRIVER SERVICE ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL ISSUE THE COMMAND AS SPECIFIED IN T.CS1
                                ;*      AND CHECK IF A CONTROLLER ERROR OCCURRED.  IF A CONTROLLER
                                ;*      ERROR OCCURRED, E.CERR WILL BE SET IN E.CONT AND
                                ;*      CONTROL WILL BE TURN OVER TO THE ROUTINE SPECIFIED BY THE
                                ;*      ADDRESS IN A.CONT.
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R2              ADDDRESS OF RK06 REGISTERS
                                ;*      R5              ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;*CALL  JSR     R0,I.ISSU
                                ;*      <ADDRESS OF ERROR RETURN>
                                ;*      RETURN
                                ;*
                                ;*      ROUTINES USED:
                                ;*      -------------
                                ;*
                                ;*              I.CCLR
                                ;*              I.STOR
                                ;*
                                ;;*****************************************************************

037240  013746  003010        I.ISSU: MOV     T.CS1,-(SP)     ;STORE COMMAND ISSUED
037244  005037  003012                CLR     T.CS2           ;CLEAR TEMPORARY CS2
037250  116537  000000  003012        MOVB    P.DRVN(R5),T.CS2 ;LOAD IN DRIVE NUMBER
037256  013762  003012  000010        MOV     T.CS2,RKCS2(R2) ;LOAD DRIVE NUMBER FOR COMMAND
037264  116537  000007  003011        MOVB    P.CS1H(R5),T.CS1+1 ;STORE BITS 8-15 OF CS1
037272  142737  177753  003011        BICB    #^C<B.CDT!B.CFMT>,T.CS1+1 ;CLEAR ALL BITS EXCEPT
                                                                ;  FORMAT AND DRIVE TYPE
037300  013762  003010  000000        MOV     T.CS1,RKCS1(R2) ;ISSUE COMMAND
037306  105762  000000        1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
037312  100375                        BPL     1$
037314  004737  037462                JSR     PC,I.STOR       ;GO STORE REGISTERS
037320  032737  100000  003010        BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCUERED
037326  001437                        BEQ     5$              ;NO, RETURN
037330  032737  001000  003012        BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
037336  001406                        BEQ     2$              ;NO, CHECK FOR OTHER CONTROLLER ERRORS
037340  052737  100000  003062        BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT FLAG
037346  004737  040166                JSR     PC,R.CONT       ;REPORT CONTROLLER ERROR
037352  000440                        BR      10$             ;RETURN

                                ;CHECK IF ANY CONTROLLER ERROR IS SET
037354  032737  024000  003010  2$:   BIT     #CTO!SPAR,T.CS1
037362  001027                        BNE     7$
037364  032737  176400  003012        BIT     #UFE!PGE!NEM!NED!UPE!WCE!DLT,T.CS2
037372  001023                        BNE     7$
037374  032737  131761  003026        BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,T.ER
037402  001017                        BNE     7$

037404  122716  000005                CMPB    #DR.CLR,(SP)    ;CHECK IF CLEAR DRIVE
```

```
 7666  037410  001003                          BNE     3$                ;NO, DO NOT SET DRIVE HARD ERROR
 7667  037412  052765  000020  000014          BIS     #DRVHRD,P.PRST(R5) ;SET HARD DRIVE ERROR
 7668  037420  004037  037156          3$:     JSR     R0,I.CCLR         ;GO ISSUE A CONTROLLER CLEAR
 7669  037424  037454                          10$                       ;ERROR RETURN
 7670  037426  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)     ;SET INTERRUPT ENABLE
 7671  037434  005726                          TST     (SP)+             ;ADJUST STACK
 7672  037436  005720                          TST     (R0)+             ;ADJUST R0 FOR NORMAL RETURN
 7673  037440  000200                          RTS     R0                ;RETURN
 7674
 7675  037442  052737  001000  003062  7$:     BIS     #E.CERR,E.CONT    ;SET CONTROLLER ERROR DURING
 7676                                                                    ;  DRIVER SERVICING
 7677  037450  004737  040166                  JSR     PC,R.CONT         ;REPORT ERROR
 7678  037454  005726                  10$:    TST     (SP)+             ;ADJUST STACK
 7679  037456  011000                          MOV     (R0),R0           ;ADJUST R0 FOR ERROR RETURN
 7680  037460  000200                          RTS     R0                ;RETURN
```

```
                                .SBTTL   *STORE RK611 UNIBUS REGISTERS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE IS CALLED BY THE RK06 DRIVER TO STORE ALL
                                ;*      RK611 REGISTER IN TEMPORARY LOCATIONS.
                                ;*
                                ;*CALL  JSR     PC,I.STOR
                                ;*      RETURN
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R2              ADDRESS OF RK611 REGISTERS
                                ;*
                                ;;*****************************************************************

037462  016237  000000  003010  I.STOR: MOV     RKCS1(R2),T.CS1 ;STORE ALL CONTROLLER REGISTERS
037470  016237  000010  003012          MOV     RKCS2(R2),T.CS2 ;  EXCEPT DATA BUFFER
037476  016237  000002  003014          MOV     RKWC(R2),T.WCR
037504  016237  000004  003016          MOV     RKBA(R2),T.BA
037512  016237  000006  003020          MOV     RKDA(R2),T.DA
037520  016237  000012  003030          MOV     RKDS(R2),T.DS
037526  016237  000014  003026          MOV     RKER(R2),T.ER
037534  016237  000016  003024          MOV     RKASOF(R2),T.ASOF
037542  016237  000020  003022          MOV     RKDCYL(R2),T.DC
037550  016237  000026  003032          MOV     RKMR1(R2),T.MR1
037556  016237  000034  003034          MOV     RKMR2(R2),T.MR2
037564  016237  000036  003036          MOV     RKMR3(R2),T.MR3
037572  016237  000030  003040          MOV     RKECPS(R2),T.POS
037600  016237  000032  003042          MOV     RKECPT(R2),T.PAT
037606  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL   *STORE CONTROLLER STATUS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE IS CALLED BY THE RK06 DRIVER AT PRIORITY 7.
                                ;*      THE FOLLOWING REGISTERS WILL BE STORED:
                                ;*
                                ;*              COMMAND AND STATUS REGISTER 2
                                ;*              WORD COUNT REGISTER
                                ;*              BUS ADDRESS REGISTER
                                ;*              DESIRED TRACK AND SECTOR
                                ;*              STATUS REGISTER
                                ;*              ERROR REGISTER
                                ;*              ATTENTION SUMMARY/OFFSET REGISTER
                                ;*              CYLINDER ADDRESS REGISTER
                                ;*
                                ;*CALL  JSR     PC,I.CSTS
                                ;*      RETURN
                                ;*
                                ;*      THIS ROUTINE ASSUMES THE FOLLOWING REGISTERS CONTAIN:
                                ;*
                                ;*
                                ;*              REGISTER                CONTENTS
                                ;*              --------                --------
                                ;*
                                ;*              R2                      RK06 BASE ADDRESS
                                ;*              R5                      ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;;*****************************************************************

037610  042765  177741  000016  I.CSTS: BIC     #177741,P.CS1(R5)       ;CLEAR ALL BITS EXCEPT FUNCTION
                                                                        ;OF LAST COMMAND ISSUED
037616  042737  000036  003010          BIC     #36,T.CS1               ;CLEAR FUNCTION OF CS1 STATUS
037624  053765  003010  000016          BIS     T.CS1,P.CS1(R5)         ;GENERATE CS1 STATUS INFORMATION
037632  016265  000010  000020  I.CST1: MOV     RKCS2(R2),P.CS2(R5)     ;STORE COMMAND AND STATUS REGISTER 2
037640  016265  000002  000022          MOV     RKWC(R2),P.WCR(R5)      ;STORE WORD COUNT REGISTER
037646  016265  000004  000024          MOV     RKBA(R2),P.BAR(R5)      ;STORE BUS ADDRESS REGISTER
037654  016265  000006  000026          MOV     RKDA(R2),P.DTS(R5)      ;STORE DESIRED TRACK AND SECTOR
037662  016265  000012  000036          MOV     RKDS(R2),P.DS(R5)       ;STORE DRIVE STATUS REGISTER
037670  016265  000014  000034          MOV     RKER(R2),P.ER(R5)       ;STORE ERROR REGISTER
037676  016265  000016  000032          MOV     RKASOF(R2),P.ASOF(R5)   ;STORE ATTENTION SUMMARY AND
                                                                        ;  OFFSET
037704  016265  000020  000030          MOV     RKDCYL(R2),P.DCYL(R5)   ;STORE CYLINDER ADDRESS
037712  000207                  RTS     PC                              ;RETURN
```

```
                                .SBTTL   *GATHER DRIVE STATUS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE WILL BE USED TO GATHER DRIVE STATUS
                                ;*      BYTE 01, 10, AND 11. IT IS ASSUMED THAT THE DRIVE
                                ;*      HAS PREVIOUSLY BEEN SEIZED.  IT RUNS AT PRIORITY 7.
                                ;*
                                ;*CALL  JSR     R0,I.STAT
                                ;*      <ADDRESS OF ERROR RETURN>
                                ;*      RETURN
                                ;*
                                ;*      THIS ROUTINE ASSUMES THE FOLLOWING REGISTERS CONTAIN:
                                ;*
                                ;*
                                ;*              REGISTER                CONTENTS
                                ;*              --------                --------
                                ;*
                                ;*              R2                      RK06 BASE ADDRESS
                                ;*              R5                      ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;*      ROUTINES USED:
                                ;*              I.ISSU
                                ;*
                                ;;*****************************************************************

037714  012762  000001  000026  I.STAT: MOV     #1,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 01
037722  112737  000001  003010          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
037730  004037  037240                  JSR     R0,I.ISSU               ;GET STATUS BYTES 01
037734  040124                          3$                              ;ERROR RETURN
037736  013765  003034  000044          MOV     T.MR2,P.A01(R5)         ;STORE STATUS BYTE 01 MESS A
037744  013765  003036  000046          MOV     T.MR3,P.B01(R5)         ;STORE STATUS BYTE 01 MESS B
037752  012762  000002  000026          MOV     #2,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 10
037760  112737  000001  003010          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
037766  004037  037240                  JSR     R0,I.ISSU               ;GET STATUS BYTES 10
037772  040124                          3$                              ;ERROR RETURN
037774  013765  003034  000050          MOV     T.MR2,P.A10(R5)         ;STORE STATUS BYTE 10 MESS A
040002  013765  003036  000052          MOV     T.MR3,P.B10(R5)         ;STORE STATUS BYTE 10 MESS B
040010  012762  000003  000026          MOV     #3,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER
                                                                        ;  FOR STATUS BYTE 11
040016  112737  000001  003010          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
040024  004037  037240                  JSR     R0,I.ISSU               ;GET STATUS BYTES 11
040030  040124                          3$                              ;ERROR RETURN
040032  013765  003034  000054          MOV     T.MR2,P.A11(R5)         ;STORE STATUS BYTE 11 MESS A
040040  013765  003036  000056          MOV     T.MR3,P.B11(R5)         ;STORE STATUS BYTE 11 MESS B
040046  005062  000026                  CLR     RKMR1(R2)               ;LOAD MAINTENANCE REGISTER 1
                                                                        ;  FOR STATUS BYTE 00
040052  112737  000001  003010          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
040060  004037  037240                  JSR     R0,I.ISSU               ;GET STATUS BYTES 00
040064  040124                          3$                              ;ERROR RETURN
040066  013765  003034  000040          MOV     T.MR2,P.A00(R5)         ;STORE STATUS BYTE 00 MESS A
040074  013765  003036  000042          MOV     T.MR3,P.B00(R5)         ;STORE STATUS BYTE 00 MESS B
040102  032737  001000  003036          BIT     #S.PAR,T.MR3            ;CHECK IF BAD PARITY DETECTED BY DRIVE
040110  001407                          BEQ     5$                      ;NO, RETURN NORMALLY
```

```
 7814  040112  052737  002000  003062          BIS     #E.CPAR,E.CONT          ;INDICATE BAD PARITY DETECTED BY DRIVE
 7815  040120  004737  040166                  JSR     PC,R.CONT               ;REPORT ERROR
 7816  040124  011000                  3$:     MOV     (R0),R0                 ;LOAD R0 FOR ERROR RETURN
 7817  040126  000200                          RTS     R0                      ;RETURN
 7818
 7819  040130  052765  001000  000014  5$:     BIS     #PBSVAL,P.PRST(R5)      ;SET PARAMETER BLOCK STATUS VALID
 7820  040136  005720                          TST     (R0)+                   ;ADJUST R0 FOR NORMAL RETURN
 7821  040140  000200                          RTS     R0                      ;RETURN
 7822
```

```
                                        .SBTTL   *COMMON DRIVER RETURNS

040142  105037  003111          R.ABNL: CLRB    INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
040146  004777  142704                  JSR     PC,@A.ABNL      ;INDICATE ABNORMAL RETURN
040152  000207                          RTS     PC              ;RETURN

040154  105037  003111          R.NORM: CLRB    INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
040160  004777  142670                  JSR     PC,@A.NORM      ;INDICATE NORMAL RETURN
040164  000207                          RTS     PC              ;RETURN

040166  105037  003111          R.CONT: CLRB    INTMSK          ;INHIBIT FUTURE DRIVE INTERRUPT REPORTING
040172  105037  003110                  CLRB    W.TIME          ;RESET WATCH DOG TIMING ON THIS DRIVE
040176  005037  003124                  CLR     W.DRV           ;CLEAR TIMING COUNT FOR THIS DRIVE
040202  004777  142652                  JSR     PC,@A.CONT      ;INDICATE CONTROLLER ERROR RETURN
040206  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL   *COMMAND INITATOR

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE WILL INITIATE ALL COMMANDS AS SPECIFIED
                                ;*      BY THE COMMAND FIELD OF THE PARAMETER BLOCK. THE FOLLOWING
                                ;*      SPECIAL COMMAND ARE ALSO EXECUTED:
                                ;*
                                ;*              RELEASE
                                ;*              CONROLLER CLEAR
                                ;*              SUBSYSTEM CLEAR
                                ;*              READ ALL DRIVE STATUS
                                ;*              READ SPECIFIED HEADER
                                ;*
                                ;*      THE ABOVE COMMANDS ARE TRANSLATED INTO A SEQUENCE OF COMMANDS
                                ;*
                                ;*CALL  JSR     PC,C.INIT
                                ;*      <ADDRESS OF PARAMETER BLOCK>
                                ;*      RETURN
                                ;*
                                ;*      FOR THE SEQUENTIAL OPERATIONS, THE DRIVER WILL LOAD THE
                                ;*      LOCATIONS, PBLKT AND INTMSK.
                                ;*
                                ;*      ROUTINES USED:
                                ;*              W.WTCH
                                ;*              I.CSTS
                                ;*              I.STAT
                                ;*              I.CCLR
                                ;*
                                ;;*****************************************************************

040210  010546                  C.INIT: MOV     R5,-(SP)        ;STORE R5 ON STACK
040212  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
040214  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
040216  010246                          MOV     R2,-(SP)        ;STORE R2 ON STACK
040220  010146                          MOV     R1,-(SP)        ;STORE R1 ON STACK
040222  010046                          MOV     R0,-(SP)        ;STORE R0 ON STACK
040224  013746  177776                  MOV     PS,-(SP)        ;STORE PSW ON STACK
040230  013737  003052  177776          MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
040236  017605  000016                  MOV     @16(SP),R5      ;STORE PARAMETER BLOCK ADDRESS
040242  062766  000002  000016          ADD     #2,16(SP)       ;ADJUST RETURN
040250  016504  000000                  MOV     P.DRVN(R5),R4   ;STORE DRIVE NUMBER
040254  042704  177770                  BIC     #^C<DRVMSK>,R4  ;MASK OUT JUNK
040260  010537  003122                  MOV     R5,PBLKT        ;LOAD PARAMETER BLOCK TABLE
040264  116437  003112  003111          MOVB    I.DRV(R4),INTMSK ;LOAD INTERRUPT MASK
040272  116437  003112  003110          MOVB    I.DRV(R4),W.TIME ;SET WATCH-DOG TIMER FLAG
040300  013737  003072  003124          MOV     W.SEC,W.DRV     ;LOAD WATCH-DOG TIME

040306  013702  003046                  MOV     RKBAS,R2        ;LOAD R2 WITH RK06 ADDRESS BASE

                                ;       RESET ALL BITS IN PROGRAM DEVICE STATUS REGISTER EXCEPT
                                ;               DRIVE IN USE
                                ;               WRITE FOR WRITE CHECK
                                ;               NO CHECK
                                ;               DROP DRIVE FROM TEST SEQUENCE
                                ;               INHIBIT BUS ADDRESS INCREMENT
```

```
 7894  040312  042765  075176  000014          BIC     #^C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
 7895
 7896  040320  010500                          MOV     R5,R0           ;STORE PARAMETER BLOCK ADDRESS
 7897  040322  062700  000016                  ADD     #P.CS1,R0       ;CALCULATE FIRST LOCATION TO BE CLEARED
 7898  040326  010501                          MOV     R5,R1           ;STORE PARAMETER BLOCK ADDRESS
 7899  040330  062701  000062                  ADD     #P.EPAT,R1      ;CALCULATE LAST LOCATION TO BE CLEARED
 7900
 7901  040334  005020                  1$:     CLR     (R0)+           ;CLEAR RETURN PARAMETER
 7902  040336  020001                          CMP     R0,R1           ;CHECK IF FINISHED
 7903  040340  101775                          BLOS    1$              ;NO, CLEAR NEXT RETURN PARAMETER
 7904  040342  105037  003104                  CLRB    I.ISRL          ;CLEAR RELEASE OR INTERRUPT ISSUED
 7905  040346  010465  000020                  MOV     R4,P.CS2(R5)    ;STORE DRIVE NUMBER
 7906  040352  005062  000026                  CLR     RKMR1(R2)       ;CLEAR RK06 MAINTENANCE REGISTER 1
 7907  040356  132765  000040  000001          BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
 7908  040364  001402                          BEQ     3$              ;NO, PROCESS
 7909  040366  000137  041102                  JMP     C.SPEC          ;JUMP TO SPECIAL COMMAND PROCESSOR
 7910
 7911  040372  122765  000107  000001  3$:     CMPB    #UNLOAD,P.CMND(R5) ;CHECK IF POSITIONING COMMAND
 7912                                                                  ;  START SPINDLE
 7913                                                                  ;  RECALIBRATE
 7914                                                                  ;  OFFSET
 7915                                                                  ;  SEEK
 7916                                                                  ;  UNLOAD
 7917
 7918  040400  101174                          BHI     25$             ;NO, DRIVE COMMAND
 7919                                                                  ;  SELECT DRIVE
 7920                                                                  ;  PACK ACKNOWLEDGE
 7921                                                                  ;  CLEAR
 7922
 7923  040402  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF DATA TRANSFER
 7924  040410  103540                          BLO     20$             ;YES, DATA TRANSFER COMMAND
 7925                                                                  ;  READ DATA
 7926                                                                  ;  WRITE DATA
 7927                                                                  ;  READ HEADER
 7928                                                                  ;  WRITE HEADER
 7929                                                                  ;  WRITE CHECK
 7930  040412  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 7931  040420  052765  000002  000014          BIS     #DRVPOS,P.PRST(R5) ;SET DRIVE POSITIONING
 7932  040426  005037  003064                  CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND
 7933  040432  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK
 7934  040440  001007                          BNE     5$              ;NO, CHECK FOR OFFSET
 7935  040442  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 7936  040450  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
 7937  040456  000431                          BR      8$              ;GO ISSUE COMMAND
 7938
 7939  040460  122765  000115  000001  5$:     CMPB    #OFFSET,P.CMND(R5) ;CHECK IF OFFSET
 7940  040466  001007                          BNE     6$              ;NO, CHECK FOR UNLOAD
 7941  040470  116565  000006  000032          MOVB    P.OFST(R5),P.ASOF(R5) ;STORE OFFSET
 7942  040476  016562  000032  000016          MOV     P.ASOF(R5),RKASOF(R2) ;LOAD OFFSET REGISTER
 7943  040504  000416                          BR      8$              ;GO ISSUE COMMAND
 7944
 7945  040506  122765  000111  000001  6$:     CMPB    #SRTSPL,P.CMND(R5) ;CHECK IF START SPINDLE
 7946  040514  001003                          BNE     7$              ;NO, CHECK IF RECAL
 7947  040516  013737  003076  003124          MOV     W.MIN,W.DRV     ;LOAD WATCH DOG TIME FOR 1 MINUTE
 7948  040524  122765  000113  000001  7$:     CMPB    #RECAL,P.CMND(R5) ;CHECK IF RECAL
 7949  040532  001003                          BNE     8$              ;NO, CONTINUE
```

```
 7950  040534  013737  003074  003124          MOV     W.8SEC,W.DRV    ;LOAD RECAL TIME FOR 8 SECONDS
 7951  040542  116565  000007  000017  8$:     MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 7952  040550  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 7953                                                                  ;  AND DRIVE TYPE
 7954  040556  116565  000001  000016          MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 7955  040564  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 7956  040572  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IN NO CHECK MODE
 7957  040600  001533                          BEQ     30$             ;NO, SKIP CLEAR OF INTERRUPT ENABLE
 7958  040602  042765  000100  000016          BIC     #IE,P.CS1(R5)   ;CLEAR INTERRUPT ENABLE
 7959  040610  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 7960  040616  004737  034560          10$:    JSR     PC,W.WTCH       ;CALL WATCH DOG TIMER
 7961  040622  016237  000000  003010          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
 7962  040630  032737  000200  003010          BIT     #RDY,T.CS1      ;WAIT FOR READY
 7963  040636  001767                          BEQ     10$
 7964  040640  032737  100000  003010          BIT     #CERR,T.CS1     ;CHECK FOR ERROR
 7965  040646  001011                          BNE     15$             ;YES, GIVE NORMAL RETURN
 7966  040650  004737  034560          11$:    JSR     PC,W.WTCH       ;CALL WATCH DOG TIMER
 7967  040654  016237  000016  003024          MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 7968  040662  133737  003111  003025          BITB    INTMSK,T.ASOF+1 ;CHECK IF INTERRUPT HAS OCCURRED
 7969  040670  001767                          BEQ     11$             ;WAIT FOR DRIVE INTERRUPT
 7970  040672  105037  003110          15$:    CLRB    W.TIME          ;RESET TIMING ON THIS DRIVE
 7971  040676  005037  003124                  CLR     W.DRV           ;CLEAR DRIVE TIMING COUNT
 7972  040702  004737  040154                  JSR     PC,R.NORM       ;INDICATE COMMAND IS FINISHED
 7973  040706  000137  042062                  JMP     C.RTRN          ;RESTORE REGISTERS
 7974
 7975  040712  016562  000010  000004  20$:    MOV     P.BALO(R5),RKBA(R2) ;LOAD BUS ADDRESS REGISTER
 7976  040720  016562  000012  000002          MOV     P.WC(R5),RKWC(R2) ;LOAD WORD COUNT REGISTER
 7977  040726  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
 7978  040734  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK NUMBER
 7979  040742  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)  ;CHECK IF WRITE CHECK COMMAND
 7980  040750  001010                          BNE     25$             ;NO, GO ISSUE THE COMMAND
 7981  040752  032765  000200  000014          BIT     #W.WCK,P.PRST(R5)  ;CHECK IF WRITE COMMAND SHOULD BE ISSUED
 7982  040760  001404                          BEQ     25$             ;NO, GO ISSUE THE COMMAND
 7983  040762  012765  000123  000016          MOV     #WRDATA,P.CS1(R5)  ;ISSUE WRITE COMMAND
 7984  040770  000406                          BR      26$             ;GO ISSUE COMMAND
 7985
 7986  040772  116565  000001  000016  25$:    MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 7987  041000  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 7988  041006  116565  000007  000017  26$:    MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 7989  041014  142765  177750  000017          BICB    #^C<B.CFMT!B.CDT!B.BA16!B.BA17>,P.CS1+1(R5) ;CLEAR ALL BI S EXCEPT
 7990                                                                  ;  FORMAT, DRIVE TYPE, AND BUS ADDRESS
 7991                                                                  ;  BITS 16-17
 7992  041022  010537  003054                  MOV     R5,O.WAIT       ;LOAD WAITING FOR COMMAND
 7993  041026  032765  100000  000014          BIT     #DTBAII,P.PRST(R5) ;CHECK IF INHIBIT BUS ADDRESS INCREMENT
 7994  041034  001403                          BEQ     27$             ;NO, LOAD CS2
 7995  041036  052765  000020  000020          BIS     #BAI,P.CS2(R5)  ;SET INHIBIT BUS ADDRESS INCREMENT
 7996  041044  016562  000020  000010  27$:    MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2
 7997  041052  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IN NO CHECK MODE
 7998  041060  001403                          BEQ     30$             ;NO, SKIP CLEAR OF INTERRUPT ENABLE
 7999  041062  042765  000100  000016          BIC     #IE,P.CS1(R5)   ;CLEAR INTERRUPT ENABLE
 8000  041070  016562  000016  000000  30$:    MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 8001  041076  000137  042062                  JMP     C.RTRN          ;RESTORE REGISTERS
 8002
 8003                                  .SBTTL   *SPECIAL COMMAND PROCESSING
 8004
 8005  041102  122765  000141  000001  C.SPEC: CMPB    #RDSTAT,P.CMND(R5) ;CHECK IF READ DRIVE STATUS
```

```
 8006  041110  001132                          BNE     10$             ;NO, PROCESS OTHER COMMANDS
 8007  041112  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR COMMAND
 8008  041120  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 8009  041126  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 8010                                                                  ;  AND DRIVE TYPE
 8011  041134  112765  000001  000016          MOVB    #DR.SEL,P.CS1(R5) ;STORE COMMAND
 8012  041142  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 8013  041150  004737  034560          2$:     JSR     PC,W.WTCH       ;CALL WATCH-DOG TIMER
 8014  041154  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REG. 1
 8015  041162  032765  000200  000016          BIT     #RDY,P.CS1(R5)  ;WAIT FOR READY
 8016  041170  001767                          BEQ     2$
 8017  041172  004737  037632                  JSR     PC,I.CST1       ;STORE CONTROLLER REGISTERS
 8018  041176  016265  000034  000040          MOV     RKMR2(R2),P.A00(R5) ;STORE STATUS BYTE 00 MESSAGE A
 8019  041204  016265  000036  000042          MOV     RKMR3(R2),P.B00(R5) ;STORE STATUS BYTE 00 MESSAGE B
 8020  041212  032765  100000  000016          BIT     #CERR,P.CS1(R5) ;CHECK IF CONTROLLER ERROR
 8021  041220  001436                          BEQ     6$              ;NO, GATHER DRIVE STATUS
 8022  041222  105037  003110                  CLRB    W.TIME          ;RESET WATCH DOG TIMING ON THIS DRIVE
 8023  041226  005037  003124                  CLR     W.DRV           ;CLEAR WATCH DOG COUNT
 8024  041232  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8025  041240  001043                          BNE     8$              ;YES, INDICATE NORMAL RETURN
 8026  041242  032765  001000  000020          BIT     #MDS,P.CS2(R5)  ;CHECK IF MULTIPLE DRIVE SELECT
 8027  041250  001043                          BNE     9$              ;YES, INDICATE CONTROLLER ERROR
 8028  041252  004037  037156                  JSR     R0,I.CCLR       ;CLEAR ERROR
 8029  041256  042062                          C.RTRN                  ;ERROR RETURN
 8030  041260  032765  010400  000020          BIT     #NED!UFE,P.CS2(R5) ;CHECK IF NON-EXISTENT DRIVE OR UNIT FIELD ERROR
 8031  041266  001007                          BNE     5$              ;REPORT ERROR
 8032  041270  032765  000001  000036          BIT     #DRA,P.DS(R5)   ;CHECK IF DRIVE AVAILIABLE
 8033  041276  001003                          BNE     5$              ;YES, REPORT ERROR
 8034  041300  052765  010000  000014          BIS     #DRVSZD,P.PRST(R5) ;INDICATE DRIVE IS SEIZED BY OTHER PORT
 8035  041306  004737  040142          5$:     JSR     PC,R.ABNL       ;INDICATE ABNORMAL RETURN
 8036  041312  000137  042062                  JMP     C.RTRN          ;RESTORE REGISTERS
 8037
 8038  041316  004037  037714          6$:     JSR     R0,I.STAT       ;GATHER DRIVE STATUS
 8039  041322  042062                          C.RTRN                  ;ERROR RETURN
 8040  041324  105037  003110                  CLRB    W.TIME          ;STOP WATCH-DOG TIMING ON DRIVE
 8041  041330  005037  003124                  CLR     W.DRV           ;RESET WATCH-DOG TIME
 8042  041334  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8043  041342  001402                          BEQ     8$              ;NO, REPORT ERROR
 8044  041344  005062  000000                  CLR     RKCS1(R2)       ;CLEAR INTERRUPT ENABLE
 8045  041350  004737  040154          8$:     JSR     PC,R.NORM       ;REPORT COMMAND COMPLETE
 8046  041354  000137  042062                  JMP     C.RTRN          ;RESTORE REGISTERS
 8047
 8048  041360  052737  100000  003062  9$:     BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT
 8049  041366  004737  040166                  JSR     PC,R.CONT       ;INDICATE CONTROLLER ERROR
 8050  041372  000137  042062                  JMP     C.RTRN
 8051
 8052  041376  122765  000140  000001  10$:    CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
 8053  041404  001040                          BNE     13$             ;NO, CHECK IF READ ALL HEADERS
 8054  041406  010537  003064                  MOV     R5,O.WAIT       ;STORE PARAMETER BLOCK ADDRESS IN
 8055                                                                  ;  WAIT FOR COMMAND
 8056  041412  052765  000010  000020          BIS     #RLS,P.CS2(R5)  ;SET RELEASE BIT
 8057  041420  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR DESELECT
 8058  041426  112737  000001  003104          MOVB    #1,I.ISRL       ;SET FLAG FOR RELEASE COMMAND
 8059  041434  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5);STORE BITS 8-15 OF CS1
 8060  041442  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 8061                                                                  ;  AND DRIVE TYPE
```

```
 8062  041450  112765  000101  000016          MOVB    #SELDRV,P.CS1(R5) ;STORE COMMAND
 8063  041456  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8064  041464  001403                          BEQ     11$               ;NO, DO NOT RESET INTERRUPT ENABLE
 8065  041466  042765  000100  000016          BIC     #IE,P.CS1(R5)     ;RESET INTERRUPT ENABLE
 8066  041474  016562  000016  000000  11$:    MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 8067  041502  000137  042062                  JMP     C.RTRN            ;RESTORE REGISTERS
 8068
 8069  041506  122765  000164  000001  13$:    CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 8070  041514  001053                          BNE     30$               ;NO, CHECK IF CONTROLLER CLEAR
 8071  041516  010537  003064                  MOV     R5,O.WAIT         ;SET WAITING FOR COMMAND COMMPLETION
 8072  041522  016537  000010  003100          MOV     P.BALO(R5),HDR.AD ;LOAD HEADER ADDRESS
 8073  041530  132765  000020  000007          BITB    #B.CFMT,P.CS1H(R5) ;CHECK IF 22 SECTOR FORMANT
 8074  041536  001404                          BEQ     14$               ;YES, LOAD 22 IN HEADER COUNT
 8075  041540  012737  000024  003102          MOV     #20.,HDR.CT       ;LOAD 20 IN SECTOR COUNT
 8076  041546  000403                          BR      22$               ;GO ISSUE READ HEADER COMMAND
 8077
 8078  041550  012737  000026  003102  14$:    MOV     #22.,HDR.CT       ;LOAD 22 IN SECTOR COUNT
 8079  041556  016562  000002  000020  22$:    MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 8080  041564  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD TRACK NUMBER
 8081  041572  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 8082  041600  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 8083  041606  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT DRIVE TYPE
 8084                                                                  ;  AND FORMAT
 8085  041614  112765  000125  000016          MOVB    #RDHEAD,P.CS1(R5) ;STORE READ HEADER COMMAND
 8086  041622  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8087  041630  001027                          BNE     34$               ;YES, INDICATE ILLEGAL DRIVER COMMAND
 8088  041632  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE READ HEADER
 8089  041640  000137  042062                  JMP     C.RTRN            ;RESTORE REGISTERS
 8090
 8091  041644  122765  000176  000001  30$:    CMPB    #CONCLR,P.CMND(R5) ;CHECK IF CONTROLLER CLEAR
 8092  041652  001012                          BNE     32$               ;NO, CHECK IF SUBSYSTEM CLEAR
 8093  041654  004037  037156                  JSR     R0,I.CCLR         ;CLEAR CONTROLLER
 8094  041660  042062                          C.RTRN                    ;ERROR RETURN
 8095  041662  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8096  041670  001472                          BEQ     40$               ;NO, INDICATE NORMAL RETURN
 8097  041672  005062  000000                  CLR     RKCS1(R2)         ;RESET INTERRUPT ENABLE
 8098  041676  000467                          BR      40$               ;INDICATE NORMAL RETURN
 8099
 8100  041700  122765  000177  000001  32$:    CMPB    #SUBCLR,P.CMND(R5) ;CHECK IF SUBSYSTEM CLEAR
 8101  041706  001406                          BEQ     36$               ;YES, CLEAR SUBSYSTEM
 8102  041710  052737  000100  003062  34$:    BIS     #E.ILLD,E.CONT    ;SET ILLEGAL DRIVER COMMAND
 8103  041716  004737  040166                  JSR     PC,R.CONT         ;REPORT ERROR
 8104  041722  000457                          BR      C.RTRN            ;RESTORE REGISTERS
 8105
 8106  041724  012762  000040  000010  36$:    MOV     #SCLR,RKCS2(R2)   ;ISSUE SUBSYSTEM CLEAR
 8107  041732  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REGISTER 1
 8108  041740  032765  100000  000016          BIT     #CERR,P.CS1(R5)   ;CLEAR IF CONTROLLER ERROR RESET
 8109  041746  001406                          BEQ     37$               ;NO, FINISH COMMAND
 8110  041750  052737  000001  003062          BIS     #BIT0,E.CONT      ;SET CLEAR SUBSYSTEM DID NOT CLEAR
 8111                                                                  ;  CONTROLLER ERROR
 8112  041756  004737  040166                  JSR     PC,R.CONT         ;REPORT ERROR
 8113  041762  000437                          BR      C.RTRN            ;RESTORE REGISTERS
 8114
 8115  041764  013746  003070          37$:    MOV     W.MILI,-(SP)      ;LOAD 16 MILI-SECOND COUNT FOR ATTENTION
 8116                                                                  ;  TO DISAPPEAR
 8117  041770  016265  000000  000016  38$:    MOV     RKCS1(R2),P.CS1(R5) ;STORE CS1
```

```
 8118  041776  032765  040000  000016          BIT     #DI,P.CS1(R5)   ;CHECK IF ATTENTIONS CLEARED
 8119  042004  001411                          BEQ     39$             ;YES, FINISH COMMAND
 8120  042006  005316                          DEC     (SP)            ;DECREMENT 16 MILISECOND COUNT
 8121  042010  001367                          BNE     38$             ;CHECK DRIVE INTERRUPT AGAIN
 8122  042012  005726                          TST     (SP)+           ;ADJUST STACK
 8123  042014  052737  000040  003062          BIS     #E.SCLR,E.CONT  ;SET SUBSYSTEM CLEAR DID NOT CLEAR
 8124                                                                  ;  DRIVE ATTENTIONS
 8125  042022  004737  040166                  JSR     PC,R.CONT       ;REPORT ERROR
 8126  042026  000415                          BR      C.RTRN          ;RESTORE REGISTER
 8127
 8128  042030  005726                  39$:    TST     (SP)+           ;ADJUST STACK
 8129  042032  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 8130  042040  001010                          BNE     C.RTRN          ;YES, RESTORE REGISTERS
 8131  042042  112737  177777  003104          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE SET
 8132  042050  012762  000100  000000          MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
 8133  042056  004737  040154          40$:    JSR     PC,R.NORM       ;INDICATE NORMAL TERMINATION
 8134
 8135  042062  012637  177776          C.RTRN: MOV     (SP)+,PS        ;RESTORE PSW
 8136  042066  012600                          MOV     (SP)+,R0        ;RESTORE R0
 8137  042070  012601                          MOV     (SP)+,R1        ;RESTORE R1
 8138  042072  012602                          MOV     (SP)+,R2        ;RESTORE R2
 8139  042074  012603                          MOV     (SP)+,R3        ;RESTORE R3
 8140  042076  012604                          MOV     (SP)+,R4        ;RESTORE R4
 8141  042100  012605                          MOV     (SP)+,R5        ;RESTORE R5
 8142  042102  000207                          RTS     PC              ;RETURN
 8143                                  .SBTTL  OCTAL TO BINARY CONVERSION ROUTINE
 8144
 8145                                  ;;******************************************************************
 8146                                  ;*
 8147                                  ;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
 8148                                  ;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL
 8149                                  ;*      IT  WILL GENERATE TWO BINARY WORDS PLACING THE LOW 16 BITS
 8150                                  ;*      ON THE STACK AND THE HIGH 16 BITS IN LOCATION $HIOCT.
 8151                                  ;*
 8152                                  ;*CALL
 8153                                  ;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
 8154                                  ;*      JSR     PC,OCTBIN
 8155                                  ;*      <ADDRESS OF ERROR RETURN>
 8156                                  ;*      RETURN
 8157                                  ;*
 8158                                  ;;******************************************************************
 8159
 8160  042104  010046                  OCTBIN: MOV     R0,-(SP)        ;SAVE R0
 8161  042106  010146                          MOV     R1,-(SP)        ;SAVE R1
 8162  042110  010246                          MOV     R2,-(SP)        ;SAVE R2
 8163  042112  016600  000010                  MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
 8164  042116  005001                          CLR     R1              ;CLEAR DATA WORDS
 8165  042120  005002                          CLR     R2
 8166  042122  112046                  2$:     MOVB    (R0)+,-(SP)     ;PICK THIS CHARACTER
 8167  042124  001423                          BEQ     3$              ;IF ZERO GET OUT
 8168  042126  121627  000054                  CMPB    (SP),#',        ;CHECK IF COMMA
 8169  042132  001420                          BEQ     3$              ;IF COMMA GET OUT
 8170  042134  122716  000060                  CMPB    #'0,(SP)        ;MAKE SURE THIS CHARACTER IS
 8171  042140  003030                          BGT     4$              ;  AN OCTAL DIGIT
 8172  042142  122716  000067                  CMPB    #'7,(SP)
 8173  042146  002425                          BLT     4$
```

```
 8174  042150  006301                        ASL     R1              ; *2
 8175  042152  006102                        ROL     R2
 8176  042154  006301                        ASL     R1              ; *4
 8177  042156  006102                        ROL     R2
 8178  042160  006301                        ASL     R1              ; *8
 8179  042162  006102                        ROL     R2
 8180  042164  042716  177770                BIC     #^C7,(SP)       ;STRIP THE ASCII JUNK
 8181  042170  062601                        ADD     (SP)+,R1        ;ADD THIS DIGIT
 8182  042172  000753                        BR      2$              ;LOOP
 8183  042174  005726                3$:     TST     (SP)+           ;CLEAN PARTIAL FROM STACK
 8184  042176  010166  000010                MOV     R1,10(SP)       ;SAVE RESULT
 8185  042202  010237  042236                MOV     R2,$HIOCT
 8186  042206  012602                        MOV     (SP)+,R2        ;RESTORE R2
 8187  042210  012601                        MOV     (SP)+,R1        ;RESTORE R1
 8188  042212  012600                        MOV     (SP)+,R0        ;RESTORE R0
 8189  042214  062716  000002                ADD     #2,(SP)         ;ADJUST RETURN
 8190  042220  000207                        RTS     PC              ;RETURN
 8191
 8192  042222  005726                4$:     TST     (SP)+           ;CLEAN UP PARTIAL FROM STACK
 8193  042224  012602                        MOV     (SP)+,R2        ;RESTORE R2
 8194  042226  012601                        MOV     (SP)+,R1        ;RESTORE R1
 8195  042230  012600                        MOV     (SP)+,R0        ;RESTORE R0
 8196  042232  013616                        MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ROUTINE ON STACK
 8197  042234  000207                        RTS     PC              ;GO PROCESS ERROR
 8198  042236  000000                $HIOCT: .WORD   0               ;HIGH ORDER BITS GO HERE
 8199                                  .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
 8200
 8201                                  ;;**************************************************************************
 8202                                  ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
 8203                                  ;*WITH A RANGE OF 0 TO 2(+33)-1.
 8204                                  ;*CALL:
 8205                                  ;*      JSP     PC,$RAND        ;;CALL THE ROUTINE
 8206                                  ;*      RETURN                  ;;RETURN HERE THE RANDOM
 8207                                  ;*                              ;;NUMBER WILL BE IN
 8208                                  ;*                              ;;$HINUM,$LONUM
 8209
 8210  042240                          $RAND:
 8211  042240  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8212  042242  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8213  042244  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8214  042246  013700  042340                MOV     $LONUM,R0       ;;SET R0 WITH LOW
 8215  042252  013701  042336                MOV     $HINUM,R1       ;;SET R1 WITH HIGH
 8216  042256  012702  177771                MOV     #-7,R2          ;;SET SHIFT COUNT
 8217  042262  006300                1$:     ASL     R0              ;;SHIFT R0 LEFT AND
 8218  042264  006101                        ROL     R1              ;;ROTATE CARRY INTO R1 AND
 8219  042266  005202                        INC     R2              ;;CHECK FOR DONE
 8220  042270  001374                        BNE     1$              ;;CONTINUE SHIFT LOOP
 8221  042272  063700  042340                ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
 8222  042276  005501                        ADC     R1              ;;PROPOGATE CARRY
 8223  042300  063701  042336                ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
 8224  042304  062700  001057                ADD     #1057,R0        ;;ADD LOW CONSTANT
 8225  042310  005501                        ADC     R1              ;;PROPOGATE CARRY
 8226  042312  062701  047401                ADD     #47401,R1       ;;ADD HIGH CONSTANT
 8227  042316  010037  042340                MOV     R0,$LONUM       ;;SAVE R0
 8228  042322  010137  042336                MOV     R1,$HINUM       ;;SAVE R1
 8229  042326  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 8230  042330  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8231  042332  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8232  042334  000207                          RTS     PC              ;;RETURN
 8233  042336  176543                $HINUM: .WORD   176543
 8234  042340  123456                $LONUM: .WORD   123456
 8235                                  .SBTTL  TYPE ROUTINE
 8236
 8237                                  ;;*****************************************************************
 8238                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 8239                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 8240                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 8241                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 8242                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 8243                                  ;*
 8244                                  ;*CALL:
 8245                                  ;*1) USING A TRAP INSTRUCTION
 8246                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 8247                                  ;*OR
 8248                                  ;*      TYPE
 8249                                  ;*      MESADR
 8250                                  ;*
 8251
 8252  042342  105737  001157  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 8253  042346  100002                  BPL     1$              ;;BR IF YES
 8254  042350  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
 8255  042352  000430                  BR      3$              ;;LEAVE
 8256  042354  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
 8257  042356  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 8258  042362  122737  000001  001350  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 8259  042370  001011                  BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 8260  042372  132737  000100  001351  BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 8261  042400  001405                  BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 8262  042402  010037  042412          MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 8263  042406  004737  044752          JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 8264  042412  000000          61$:    .WORD   0               ;;MESSAGE ADDRESS
 8265  042414  132737  000040  001351  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 8266  042422  001003                  BNE     60$             ;;YES,SKIP TYPE OUT
 8267  042424  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 8268  042426  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 8269  042430  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 8270  042432  012600          60$:    MOV     (SP)+,R0        ;;RESTORE R0
 8271  042434  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 8272  042440  000002                  RTI                     ;;RETURN
 8273  042442  122716  000011  4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 8274  042446  001430                  BEQ     8$
 8275  042450  122716  000200          CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 8276  042454  001006                  BNE     5$
 8277  042456  005726                  TST     (SP)+           ;;POP  <CR><LF> EQUIV
 8278  042460  104401                  TYPE                    ;;TYPE A CR AND LF
 8279  042462  001325                  $CRLF
 8280  042464  105037  042620          CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 8281  042470  000755                  BR      2$              ;;GET NEXT CHARACTER
 8282  042472  004737  042554  5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 8283  042476  123726  001156  6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
 8284  042502  001350                  BNE     2$              ;;IF NO GO GET NEXT CHAR.
 8285  042504  013746  001154          MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
```

```
 8286
 8287  042510  105366  000001          7$:     DECB    1(SP)           ;;AND THE NULL CHAR.
 8288  042514  002770                          BLT     6$              ;;DOES A NULL NEED TO BE TYPED?
 8289  042516  004737  042554                  JSR     PC,$TYPEC       ;;BR IF NO--GO POP THE NULL OFF OF STACK
 8290  042522  105337  042620                  DECB    $CHARCNT        ;;GO TYPE A NULL
 8291  042526  000770                          BR      7$              ;;DO NOT COUNT AS A COUNT
 8292                                                                  ;;LOOP
 8293                                  ;HORIZONTAL TAB PROCESSOR
 8294
 8295  042530  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 8296  042534  004737  042554          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 8297  042540  132737  000007  042620          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 8298  042546  001372                          BNE     9$              ;;TAB STOP
 8299  042550  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
 8300  042552  000724                          BR      2$              ;;GET NEXT CHARACTER
 8301  042554  105777  136370          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 8302  042560  100375                          BPL     $TYPEC
 8303  042562  116677  000002  136362          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 8304  042570  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 8305  042576  001003                          BNE     1$              ;;BRANCH IF NO
 8306  042600  105037  042620                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 8307  042604  000406                          BR      $TYPEX          ;;EXIT
 8308  042606  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 8309  042614  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
 8310  042616  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
 8311  042620  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
 8312  042622  000207          $TYPEX: RTS     PC
 8313
 8314
 8315
 8316
 8317                          ;;*****************************************************************
 8318                          .SBTTL  DOUBLE-PRECISION MULTIPLY SUBROUTINE
 8319                          ;*      SUBROUTINE TO MULTIPLY TWO DOUBLE PRECISION INTEGERS
 8320                          ;*      USES ALL REGISTERS (R0-R5)
 8321                          ;*
 8322                          ;*      ENTER WITH   JSR   PC,M.DPIM
 8323                          ;*                   MULTIPLIER IN R2-R3
 8324                          ;*                   MULTIPLICAND IN R4-R5
 8325                          ;*                   PRODUCT RETURNED IN R0-R1-R2-R3
 8326                          ;*
 8327                          ;;*****************************************************************
 8328  042624  005000          M.DPIM: CLR     R0              ;CLEAR HI ORDER WORDS
 8329  042626  005001                  CLR     R1
 8330  042630  012746  000041          MOV     #41,-(SP)       ;MOVE 33 (DEC) TO COUNTER
 8331  042634  006000          M.DP01: ROR     R0
 8332  042636  006001                  ROR     R1
 8333  042640  006002                  ROR     R2              ;SHIFT TO ADD
 8334  042642  006003                  ROR     R3
 8335  042644  103003                  BCC     M.DP02          ;NO CARRY NO ADD
 8336  042646  060501                  ADD     R5,R1
 8337  042650  005500                  ADC     R0              :ADD DOUBLE PRECISION TO OBTAIN NEW PARTIAL
 8338  042652  060400                  ADD     R4,R0           .   PRODUCT
 8339  042654  005316          M.DP02: DEC     @SP               ECREMENT COUNTER
 8340  042656  001366                  BNE     M.DP01
 8341  042660  005726                  TST     (SP)+           ;REMOVE THE COUNTER
```

```
 8342  042662  000207                          RTS     PC
 8343
 8344
 8345                                  ;;*****************************************************************
 8346                                  .SBTTL  DOUBLE-PRECISION DIVIDE SUBROUTINE
 8347                                  ;*              SUBROUTINE TO PERFORM DOUBLE-PRECISION INTEGER DIVISION
 8348                                  ;*              USES ALL REGISTERS (R0-R5)
 8349                                  ;*
 8350                                  ;*              ENTER WITH  JSR   PC,M.DPID
 8351                                  ;*                      DIVIDEND IN R0-R1-R2-R3
 8352                                  ;*                      DIVISOR IN R4-R5
 8353                                  ;*                      REMAINDER RETURNED IN R0-R1
 8354                                  ;*                      QUOTIENT RETURNED IN R2-R3
 8355                                  ;*
 8356                                  ;;*****************************************************************
 8357  042664  012746  000040         M.DPID: MOV     #40,-(SP)       ;COUNTER FOR DIVISION CYCLES
 8358  042670  010446                          MOV     R4,-(SP)        ;HI ORDER
 8359  042672  010546                          MOV     R5,-(SP)        ;LO ORDER DIVISOR TO THE STACK
 8360  042674  005466  000002                  NEG     2(SP)           ;FORM NEGATIVE
 8361  042700  005416                          NEG     @SP             ;   VERSION OF THE DIVISOR
 8362  042702  005666  000002                  SBC     2(SP)
 8363  042706  061601                          ADD     @SP,R1
 8364  042710  005500                          ADC     R0              ;PERFORM THE INITIAL SUBTRACTION
 8365  042712  066600  000002                  ADD     2(SP),R0
 8366  042716  103445                          BCS     M.DP50          ;IF CARRY THEN OVERFLOW HAS OCCURRED
 8367  042720  005046                          CLR     -(SP)           ;THIS IS A LONGER LASTING CARRY BIT
 8368  042722  006103                  M.DP40: ROL     R3
 8369  042724  006102                          ROL     R2
 8370  042726  006101                          ROL     R1
 8371  042730  006100                          ROL     R0
 8372  042732  005716                          TST     @SP             ;TEST "CARRY INDICATOR"
 8373  042734  001410                          BEQ     M.DP41          ;IF NO "CARRY" THEN ADD ELSE SUBTRACT
 8374  042736  005016                          CLR     @SP             ;CLEAR UP FOR NEXT TIME
 8375  042740  066601  000002                  ADD     2(SP),R1
 8376  042744  005500                          ADC     R0              ;ADD -(DIVISOR)
 8377  042746  005516                          ADC     @SP     ; I     ;SET "CARRY"
 8378  042750  066600  000004                  ADD     4(SP),R0 ;<
 8379  042754  000404                          BR      M.DP42
 8380  042756  060501                  M.DP41: ADD     R5,R1
 8381  042760  005500                          ADC     R0              ;ADD +(DIVISOR)
 8382  042762  005516                          ADC     @SP     ; I     ;SET "CARRY"
 8383  042764  060400                          ADD     R4,R0   ;<
 8384  042766  005516                  M.DP42: ADC     @SP             ;SET "CARRY"
 8385  042770  005716                          TST     @SP             ;TEST THE UPDATE INDICATOR
 8386  042772  001401                          BEQ     .+4     ;>      ;IF ZERO FORGET IT
 8387  042774  005203                          INC     R3      ; I     ;NO CARRY POSSIBLE HERE
 8388  042776  005366  000006                  DEC     6(SP)   ;<      ;DECREMENT COUNTER
 8389  043002  003347                          BGT     M.DP40          ;BR IF MORE TO DO
 8390  043004  006003                          ROR     R3
 8391  043006  103404                          BCS     M.DP44
 8392  043010  060501                          ADD     R5,R1
 8393  043012  005500                          ADC     R0
 8394  043014  060400                          ADD     R4,R0
 8395  043016  000241                          CLC
 8396  043020  006103                  M.DP44: ROL     R3
 8397  043022  062706  000010                  ADD     #10,SP          ;ADJUST STACK BY 4 WORDS
```

```
 8398  043026  000242                          CLV
 8399  043030  000207                          RTS     PC
 8400  043032  062706  000006          M.DP50: ADD     #6,SP
 8401  043036  000262                          SEV
 8402  043040  000207                          RTS     PC
 8403
 8404
 8405
 8406                                  .SBTTL  DOUBLE LENGTH BINARY TO OCTAL ASCII CONVERT ROUTINE
 8407
 8408                                  ;;**********************************************************************
 8409                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT UNSIGNED BINARY NUMBER TO AN
 8410                                  ;*UNSIGNED OCTAL ASCIZ NUMBER.
 8411                                  ;*CALL
 8412                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 8413                                  ;*      JSR     PC,@#$DB2O      ;;CALL THE ROUTINE
 8414                                  ;*      RETURN                  ;;THE ADDRESS OF THE FIRST ASCIZ CHAR. IS ON THE STACK
 8415
 8416
 8417  043042  104407                  $DB2O:  SAVREG                  ;;SAVE ALL REGISTERS
 8418  043044  016601  000002                  MOV     2(SP),R1        ;;PICKUP THE POINTER TO LOW WORD
 8419  043050  012705  043161                  MOV     #$OCTVL+13.,R5  ;;POINTER TO DATA TABLE
 8420  043054  012704  000014                  MOV     #12.,R4         ;;DO ELEVEN CHARACTERS
 8421  043060  012703  177770                  MOV     #^C7,R3         ;;MASK
 8422  043064  012100                          MOV     (R1)+,R0        ;;LOWER WORD
 8423  043066  012101                          MOV     (R1)+,R1        ;;HIGH WORD
 8424  043070  005002                          CLR     R2              ;;TERMINATOR
 8425  043072  110245                  1$:     MOVB    R2,-(R5)        ;;PUT CHARACTER IN DATA TABLE
 8426  043074  010002                          MOV     R0,R2           ;;GET THIS DIGIT
 8427  043076  005304                          DEC     R4              ;;COUNT THIS CHARACTER
 8428  043100  003007                          BGT     3$              ;;BR IF NOT THE LAST DIGIT
 8429  043102  001405                          BEQ     2$              ;;BR IF IT IS THE LAST DIGIT
 8430  043104  005205                          INC     R5              ;;ALL DIGITS DONE-ADJUST POINTER FOR FIRST
 8431  043106  010566  000002                  MOV     R5,2(SP)        ;;ASCIZ CHAR. & PUT IT ON THE STACK
 8432  043112  104410                          RESREG                  ;;RESTORE ALL REGISTERS
 8433  043114  000207                          RTS     PC              ;;RETURN TO USER
 8434  043116  006203                  2$:     ASR     R3              ;;POSITION THE MASK FOR THE LAST DIGIT
 8435  043120  006001                  3$:     ROR     R1              ;;POSITION THE BINARY NUMBER FOR
 8436  043122  006000                          ROR     R0              ;;      THE NEXT OCTAL DIGIT
 8437  043124  006001                          ROR     R1
 8438  043126  006000                          ROR     R0
 8439  043130  006001                          ROR     R1
 8440  043132  006000                          ROR     R0
 8441  043134  040302                          BIC     R3,R2           ;;MASK OUT ALL JUNK
 8442  043136  062702  000060                  ADD     #'0,R2          ;;MAKE THIS CHAR. ASCII
 8443  043142  000753                          BR      1$              ;;GO PUT IT IN THE DATA TABLE
 8444  043144  000016                  $OCTVL: .BLKB   14.              ;;RESERVE DATA TABLE
 8445                                  .SBTTL  DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINE
 8446
 8447                                  ;;**********************************************************************
 8448                                  ;*THIS ROUTINE WILL CONVERT A 32-BIT BINARY NUMBER TO AN UNSIGNED
 8449                                  ;*DECIMAL (ASCII) NUMBER. THE SIGN OF THE BINARY NUMBER MUST BE
 8450                                  ;*POSITIVE.
 8451                                  ;*CALL
 8452                                  ;*      MOV     #PNTR,-(SP)     ;;POINTER TO LOW WORD OF BINARY NUMBER
 8453                                  ;*      JSR     PC,@#$DB2D
```

```
8454                                    ;*      RETURN                  ;;THE FIRST ADDRESS OF ASCIZ
8455                                                                    ;;IS ON THE STACK
8456
8457
8458  043162  104407                    $DB2D:  SAVREG                  ;;SAVE REGISTERS
8459  043164  016602  000002                    MOV     2(SP),R2        ;;PICKUP THE DATA POINTER
8460  043170  012700  043342                    MOV     #$DECVL,R0      ;;GET ADDRESS OF "$DECVL" STRING
8461  043174  010066  000002                    MOV     R0,2(SP)        ;;PUT ADDRESS OF ASCIZ STRING ON STACK
8462  043200  012201                            MOV     (R2)+,R1        ;;PICKUP THE BINARY NUMBER
8463  043202  012202                            MOV     (R2)+,R2
8464  043204  012737  000012  043260           MOV     #10.,4$         ;;SET UP TO DO 10 CONVERSIONS
8465  043212  012704  043272                    MOV     #$TNPWR,R4      ;;ADDRESS OF TEN POWER
8466  043216  012705  043274                    MOV     #$TNPWR+2,R5
8467  043222  005003                    1$:     CLR     R3              ;;CLEAR PARTIAL
8468  043224  161401                    2$:     SUB     (R4),R1         ;;SUBTRACT TEN POWER
8469  043226  005602                            SBC     R2
8470  043230  161502                            SUB     (R5),R2
8471  043232  002402                            BLT     3$              ;;BR IF TEN POWER TO LARGE
8472  043234  005203                            INC     R3              ;;ADD 1 TO PARTIAL
8473  043236  000772                            BR      2$              ;;LOOP
8474  043240  062401                    3$:     ADD     (R4)+,R1        ;;RESTORE SUBTRACTED VALUE
8475  043242  005502                            ADC     R2
8476  043244  062402                            ADD     (R4)+,R2
8477  043246  022525                            CMP     (R5)+,(R5)+     ;;MOVE TO NEXT TEN POWER
8478  043250  052703  000060                    BIS     #'0,R3          ;;CHANGE PARTIAL TO ASCII
8479  043254  110320                            MOVB    R3,(R0)+        ;;SAVE IT
8480  043256  005327                            DEC     (PC)+           ;;DONE?
8481  043260  000000                    4$:     .WORD   0
8482  043262  001357                            BNE     1$              ;;BR IF NO
8483  043264  105020                            CLRB    (R0)+           ;;TERMINATOR
8484  043266  104410                            RESREG                  ;;RESTORE REGISTERS
8485  043270  000207                            RTS     PC              ;;RETURN
8486  043272  145000                    $TNPWR: 145000                  ;;1.0E09
8487  043274  035632                            35632
8488  043276  160400                            160400                  ;;1.0E08
8489  043300  002765                            2765
8490  043302  113200                            113200                  ;;1.0E07
8491  043304  000230                            230
8492  043306  041100                            041100                  ;;1.0E06
8493  043310  000017                            17
8494  043312  103240                            103240                  ;;1.0E05
8495  043314  000001                            1
8496  043316  023420                            23420                   ;;1.0E04
8497  043320  000000                            0
8498  043322  001750                            1750                    ;;1.0E03
8499  043324  000000                            0
8500  043326  000144                            144                     ;;1.0E02
8501  043330  000000                            0
8502  043332  000012                            12                      ;;1.0E01
8503  043334  000000                            0
8504  043336  000001                            1                       ;;1.0E00
8505  043340  000000                            0
8506  043342  000014                    $DECVL: .BLKB   12.             ;;RESERVE STORAGE FOR ASCIZ STRING
8507                                    .SBTTL  TYPE NUMERICAL ASCIZ STRING SUPPRESS LEADING ZEROS
8508
8509                                    ;;**************************************************************
```

```
                                ;*THIS ROUTINE IS USED TO TYPE AN ASCIZ NUMBER SUPPRESSING THE
                                ;*LEADING NUMBERS.
                                ;*CALL
                                ;*      MOV     #NUMADR,-(SP)   ;;FIRST ADDRESS OF ASCIZ STRING
                                ;*      JSR     PC,@#$SUPRS

043356  010046                  $SUPRS: MOV     R0,-(SP)        ;;SAVE R0
043360  016600  000004                  MOV     4(SP),R0        ;;PICKUP THE POINTER
043364  105710                  1$:     TSTB    (R0)            ;;TERMINATEOR?
043366  001403                          BEQ     2$              ;;BR IF YES
043370  122720  000060                  CMPB    #'0,(R0)+       ;;IS THIS AN ASCII "0" ?
043374  001773                          BEQ     1$              ;;BR IF YES
043376  005300                  2$:     DEC     R0              ;;BACKUP BY "1"
043400  010037  043406                  MOV     R0,3$           ;;SAVE FOR TYPING
043404  104401                          TYPE                    ;;GO TYPE
043406  000000                  3$:     .WORD   0               ;;ASCIZ POINTER GOES HERE
043410  012600                          MOV     (SP)+,R0        ;;RESTORE R0
043412  012616                          MOV     (SP)+,(SP)      ;;RESTORE THE STACK
043414  000207                          RTS     PC              ;;RETURN
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;******************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP        ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

043416  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
043422  116637  000001  043641          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
043430  112637  043643                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
043434  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
043440  000406                          BR      $TYPON
043442  112737  000001  043641  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
043450  112737  000006  043643          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
043456  112737  000005  043640  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
043464  010346                          MOV     R3,-(SP)        ;;SAVE R3
043466  010446                          MOV     R4,-(SP)        ;;SAVE R4
043470  010546                          MOV     R5,-(SP)        ;;SAVE R5
```

```
 8566  043472  113704  043643                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 8567  043476  005404                        NEG     R4
 8568  043500  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 8569  043504  110437  043642                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 8570  043510  113704  043641                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 8571  043514  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 8572  043520  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
 8573  043522  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 8574  043524  000404                        BR      3$              ;;GO DO MSB
 8575  043526  006105                2$:     ROL     R5              ;;FORM THIS DIGIT
 8576  043530  006105                        ROL     R5
 8577  043532  006105                        ROL     R5
 8578  043534  010503                        MOV     R5,R3
 8579  043536  006103                3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 8580  043540  105337  043642                DECB    $OMODE          ;;TYPE THIS DIGIT?
 8581  043544  100016                        BPL     7$              ;;BR IF NO
 8582  043546  042703  177770                BIC     #177770,R3      ;;GET RID OF JUNK
 8583  043552  001002                        BNE     4$              ;;TEST FOR 0
 8584  043554  005704                        TST     R4              ;;SUPPRESS THIS 0?
 8585  043556  001403                        BEQ     5$              ;;BR IF YES
 8586  043560  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 8587  043562  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 8588  043566  052703  000040          5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 8589  043572  110337  043636                MOVB    R3,8$           ;;SAVE FOR TYPING
 8590  043576  104401  043636                TYPE    ,8$             ;;GO TYPE THIS DIGIT
 8591  043602  105337  043640          7$:   DECB    $OCNT           ;;COUNT BY 1
 8592  043606  003347                        BGT     2$              ;;BR IF MORE TO DO
 8593  043610  002402                        BLT     6$              ;;BR IF DONE
 8594  043612  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 8595  043614  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 8596  043616  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 8597  043620  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 8598  043622  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 8599  043624  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 8600  043632  012616                        MOV     (SP)+,(SP)
 8601  043634  000002                        RTI                     ;;RETURN
 8602  043636     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 8603  043637     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 8604  043640     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 8605  043641     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 8606  043642  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 8607                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 8608
 8609                                ;;**************************************************************************
 8610                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 8611                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 8612                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 8613                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 8614                                ;*REPLACED WITH SPACES.
 8615                                ;*CALL:
 8616                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 8617                                ;*      TYPDS                   ;;GO TO THE ROUTINE
 8618
 8619  043644                        $TYPDS:
 8620  043644  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8621  043646  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 8622  043650  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8623  043652  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8624  043654  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 8625  043656  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 8626  043662  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 8627  043666  100004                          BPL     1$              ;;BR IF INPUT IS POS.
 8628  043670  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 8629  043672  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 8630  043700  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 8631  043702  012703  044060                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 8632  043706  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 8633  043712  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 8634  043714  016001  044050                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 8635  043720  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 8636  043722  002402                          BLT     4$              ;;BR IF DONE
 8637  043724  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 8638  043726  000774                          BR      3$
 8639  043730  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 8640  043732  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
 8641  043734  001002                          BNE     5$              ;;FALL THROUGH IF 0
 8642  043736  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 8643  043740  100407                          BMI     7$              ;;BR IF YES
 8644  043742  106316                  5$:     ASLB    (SP)            ;;MSD?
 8645  043744  103003                          BCC     6$              ;;BR IF NO
 8646  043746  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 8647  043754  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 8648  043760  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 8649  043764  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 8650  043766  005720                          TST     (R0)+           ;;JUST INCREMENTING
 8651  043770  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
 8652  043774  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
 8653  043776  003002                          BGT     8$              ;;GO TO EXIT
 8654  044000  010502                          MOV     R5,R2           ;;GET THE LSD
 8655  044002  000764                          BR      6$              ;;GO CHANGE TO ASCII
 8656  044004  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 8657  044006  100003                          BPL     9$              ;;BR IF NO
 8658  044010  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 8659  044016  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 8660  044020  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 8661  044022  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 8662  044024  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 8663  044026  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8664  044030  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8665  044032  104401  044060                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 8666  044036  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 8667  044044  012616                          MOV     (SP)+,(SP)
 8668  044046  000002                          RTI                     ;;RETURN TO USER
 8669  044050  023420                  $DTBL:  10000.
 8670  044052  001750                          1000.
 8671  044054  000144                          100.
 8672  044056  000012                          10.
 8673  044060  000004                  $DBLK:  .BLKW   4
 8674                                  .SBTTL  ERROR HANDLER ROUTINE
 8675
 8676                                  ;;**********************************************************************
 8677                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
```

ERROR HANDLER ROUTINE SEQ 0168

```
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO TYPERR ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

044070                                  $ERROR:
044070  105237  001103                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
044074  001775                                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
044076  013777  001102  135036                  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
044104  032777  002000  135026                  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
044112  001402                                  BEQ     1$              ;;NO - SKIP
044114  104401  001320                          TYPE    ,$BELL          ;;RING BELL
044120  005237  001112                  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
044124  011637  001116                          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
044130  162737  000002  001116                  SUB     #2,$ERRPC
044136  117737  134754  001114                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
044144  032777  020000  134766                  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
044152  001004                                  BNE     20$             ;;SKIP TYPEOUTS
044154  004737  030330                          JSR     PC,TYPERR       ;;GO TO USER ERROR ROUTINE
044160  104401  001325                          TYPE    ,$CRLF
044164                                  20$:
044164  122737  000001  001350                  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
044172  001007                                  BNE     2$              ;;NO,SKIP APT ERROR REPORT
044174  113737  001114  044206                  MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
044202  004737  044762                          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
044206     000                          21$:    .BYTE   0
044207     000                                  .BYTE   0
044210  000777                          22$:    BR      22$             ;;APT ERROR LOOP
044212  005777  134722                  2$:     TST     @SWR            ;;HALT ON ERROR
044216  100001                                  BPL     3$              ;;SKIP IF CONTINUE
044220  000000                                  HALT                    ;;HALT ON ERROR!
044222  032777  001000  134710          3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
044230  001402                                  BEQ     4$              ;;BR IF NO
044232  013716  001110                          MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
044236  005737  001316                  4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
044242  001402                                  BEQ     5$              ;;BR IF NONE
044244  013716  001316                          MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
044250                                  5$:
044250  022737  017234  000042                  CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
044256  001001                                  BNE     6$              ;;BRANCH IF NO
044260  000000                                  HALT                    ;;YES
044262                                  6$:
044262  000002.                                 RTI                     ;;RETURN
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;**************************************************************
                                        .ENABL  LSB

                                        .DSABL  LSB
```

```
                                        ;;*****************************************************************
                                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*CALL:
                                        ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                        ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                        ;
044264  011646                  $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
044266  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
044274  105777  134644          1$:     TSTB    @$TKS           ;;WAIT FOR
044300  100375                          BPL     1$              ;;A CHARACTER
044302  117766  134640  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
044310  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
044316  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
044324  001013                          BNE     3$              ;;BRANCH IF NO
044326  105777  134612          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
044332  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
044334  117746  134606                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
044340  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
044344  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
044350  001366                          BNE     2$              ;;IF NOT DISCARD IT
044352  000750                          BR      1$              ;;YES, RESUME
044354  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
044362  002407                          BLT     4$              ;;BRANCH IF YES
044364  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
044372  003003                          BGT     4$              ;;BRANCH IF YES
044374  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
044402  000002                  4$:     RTI                     ;;GO BACK TO USER
044404  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
044411     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
044416  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
044424  020075     000
044427     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
044434  036440  000040


                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;POWER DOWN ROUTINE
044440  012737  044452  000024  $PWRDN: MOV     #$PWRUP,PWRVEC  ;SET VECTOR FOR POWER UP
044446  000000                          HALT                    ;HANG UP
044450  000776                          BR      .-2

                                        ;POWER UP ROUTINE
044452  005037  044524          $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR TTY TO COME UP
044456  005237  044524          4$:     INC     $PWRCT
044462  001375                          BNE     4$
044464  012737  044440  000024          MOV     #$PWRDN,PWRVEC  ;SET VECTOR FOR POWER DOWN
044472  012737  000340  000026          MOV     #PR7,PWRVEC+2   ;RE-ESTABLISH POWER AND TRAP PRIORITIES
044500  012737  000340  000036          MOV     #PR7,TRAPVEC+2
044506  012706  001100                  MOV     #STACK,SP       ;RE-INITIALIZE THE STACK
044512  104401  044526                  TYPE    ,PWRMSG         ;TYPE "POWER FAILED"
044516  000005                          RESET                   ;CLEAR THE UNIBUS
044520  000177  134362                  JMP     @$LPADR         ;RESTART THE CURRENT TEST
```

```
 8790  044524  000000                  $PWRCT: .WORD   0
 8791  044526  005015  047520  042527  PWRMSG: .ASCIZ  <15><12>/POWER FAILED/<15><12>
 8792  044534  020122  040506  046111
 8793  044542  042105  005015     000
 8794          044550                          .EVEN
 8795
 8796
 8797
 8798                                  ;;*****************************************************************
 8799                                  ;* SCOPE1 - INTERNAL SCOPE ON ERROR ROUTINE
 8800                                  ;*CALLED BY "SCOPER"
 8801                                  ;;*****************************************************************
 8802  044550  105737  001103          SCOPE1: TSTB    $ERFLG          ;SEE IF AN ERROR HAS OCCURRED
 8803  044554  00140o                          BEQ     6$              ;BR IF NOT
 8804  044556  032777  001000  134354          BIT     #BIT9,@SWR      ;SEE IF LOOP ON ERROR DESIRED
 8805  044564  001402                          BEQ     6$              ;BR IF NOT
 8806  044566  013716  001110                  MOV     $LPERR,(SP)     ;SET ERROR LOOP ADDRESS ON STACK
 8807  044572  000002                  6$:     RTI                     ;RETURN
 8808
 8809
 8810
 8811                                  .SBTTL  SCOPE HANDLER ROUTINE
 8812
 8813                                  ;;*****************************************************************
 8814                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
 8815                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
 8816                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
 8817                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 8818                                  ;*SW14=1         LOOP ON TEST
 8819                                  ;*SW09=1         LOOP ON ERROR
 8820                                  ;*CALL
 8821                                  ;*      SCOPE           ;;SCOPE-IOT
 8822
 8823  044574                          $SCOPE:
 8824  044574  032777  040000  134336  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
 8825  044602  001052                          BNE     $OVER           ;;YES IF SW14=1
 8826                                  ;#####START OF CODE FOR THE XOR TESTER#####
 8827  044604  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
 8828                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
 8829  044606  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
 8830  044612  012737  044632  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
 8831  044620  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
 8832  044624  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 8833  044630  000421                          BR      $SVLAD          ;;GO TO THE NEXT TEST
 8834  044632  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
 8835  044634  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
 8836  044640  000407                          BR      7$              ;;LOOP ON THE PRESENT TEST
 8837  044642                          6$:;#####END OF CODE FOR THE XOR TESTER#####
 8838  044642  105737  001103          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
 8839  044646  001412                          BEQ     $SVLAD          ;;BR IF NO
 8840  044650  032777  001000  134262          BIT     #BIT09,@SWR     ;;LOOP ON ERR  ?
 8841  044656  001404                          BEQ     4$              ;;BR IF NO
 8842  044660  013737  001110  001106  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
 8843  044666  000420                          BR      $OVER
 8844  044670  105037  001103          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
 8845  044674  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
```

```
 8846  044700  113737  001102  001334          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
 8847  044706  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 8848  044712  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 8849  044716  005037  001316                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 8850  044722  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 8851  044730  013777  001102  134204  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 8852  044736  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 8853  044742  000002                          RTI                     ;;FIXES PS
 8854                                  .SBTTL  APT COMMUNICATIONS ROUTINE
 8855
 8856                                  ;;**************************************************************************
 8857  044744  112737  000001  045210  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
 8858  044752  112737  000001  045206  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
 8859  044760  000403                          BR      $ATYC
 8860  044762  112737  000001  045210  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
 8861  044770                          $ATYC:
 8862  044770  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8863  044772  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8864  044774  105737  045206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
 8865  045000  001450                          BEQ     5$              ;;IF NOT:  BR
 8866  045002  122737  000001  001350          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
 8867  045010  001031                          BNE     3$              ;;IF NOT:  BR
 8868  045012  132737  000100  001351          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 8869  045020  001425                          BEQ     3$              ;;IF NOT:  BR
 8870  045022  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
 8871  045026  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 8872  045034  005737  001330          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
 8873  045040  001375                          BNE     1$              ;;IF NOT:  WAIT
 8874  045042  010037  001344                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
 8875  045046  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
 8876  045050  001376                          BNE     2$
 8877  045052  163700  001344                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
 8878  045056  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
 8879  045060  010037  001346                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
 8880  045064  012737  000004  001330          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
 8881  045072  000413                          BR      5$
 8882  045074  017637  000004  045120  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
 8883  045102  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
 8884  045110  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
 8885  045114  004737  042342                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
 8886  045120  000000                  4$:     .WORD   0
 8887  045122                          5$:
 8888  045122  105737  045210          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
 8889  045126  001416                          BEQ     12$             ;;IF NOT:  BR
 8890  045130  005737  001350                  TST     $ENV            ;;RUNNING UNDER APT?
 8891  045134  001413                          BEQ     12$             ;;IF NOT:  BR
 8892  045136  005737  001330          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
 8893  045142  001375                          BNE     11$             ;;IF NOT:  WAIT
 8894  045144  017637  000004  001332          MOV     @4(SP),$FATAL   ;;GET ERROR #
 8895  045152  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
 8896  045160  005237  001330                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
 8897  045164  105037  045210          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
 8898  045170  105037  045207                  CLRB    $LFLG           ;;CLEAR LOG FLAG
 8899  045174  105037  045206                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
 8900  045200  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8901  045202  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
 8902  045204  000207                          RTS     PC              ;;RETURN
 8903  045206     000                  $MFLG:  .BYTE   0               ;;MESSG. FLAG
 8904  045207     000                  $LFLG:  .BYTE   0               ;;LOG FLAG
 8905  045210     000                  $FFLG:  .BYTE   0               ;;FATAL FLAG
 8906           045212                          .EVEN
 8907           000200                  APTSIZE=200
 8908           000001                  APTENV=001
 8909           000100                  APTSPOOL=100
 8910           000040                  APTCSUP=040
 8911                                   .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
 8912
 8913                                   ;;**************************************************************
 8914                                   ;*SAVE R0-R5
 8915                                   ;*CALL:
 8916                                   ;*      SAVREG
 8917                                   ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
 8918                                   ;*
 8919                                   ;*TOP---(+16)
 8920                                   ;* +2---(+18)
 8921                                   ;* +4---R5
 8922                                   ;* +6---R4
 8923                                   ;* +8---R3
 8924                                   ;*+10---R2
 8925                                   ;*+12---R1
 8926                                   ;*+14---R0
 8927
 8928  045212                           $SAVREG:
 8929  045212  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 8930  045214  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 8931  045216  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 8932  045220  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 8933  045222  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 8934  045224  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 8935  045226  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
 8936  045232  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
 8937  045236  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
 8938  045242  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
 8939  045246  000002                          RTI
 8940
 8941                                   ;*RESTORE R0-R5
 8942                                   ;*CALL:
 8943                                   ;*      RESREG
 8944  045250                           $RESREG:
 8945  045250  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
 8946  045254  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
 8947  045260  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
 8948  045264  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
 8949  045270  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
 8950  045272  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
 8951  045274  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 8952  045276  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 8953  045300  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 8954  045302  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 8955  045304  000002                          RTI
 8956                                   .SBTTL  TRAP DECODER
 8957
```

```
                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

045306  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
045310  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
045314  005740                          TST     -(R0)           ;;BACKUP BY 2
045316  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
045320  006300                          ASL     R0              ;;POSITION FOR INDEXING
045322  016000  045342                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
045326  000200                          RTS     R0              ;;GO TO ROUTINE


                                ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

045330  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
045332  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
045340  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE "TRAP" INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
045342  045330                  $TRPAD: .WORD   $TRAP2
045344  042342                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
045346  043442                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
045350  043416                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
045352  043456                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
045354  043644                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)


045356  044264                          $RDCHR  ;;CALL=RDCHR    TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
045360  045212                          $SAVREG ;;CALL=SAVREG   TRAP+7(104407)  SAVE R0-R5 ROUTINE
045362  045250                          $RESREG ;;CALL=RESREG   TRAP+10(104410) RESTORE R0-R5 ROUTINE
045364  044550                          SCOPE1  ;;CALL=SCOPER   TRAP+11(104411) INTERNAL LOOP ON ERROR ROUTINE

045366  020040  020040  042524  ISTMSG: .ASCIZ  /    TEST /
045374  052123  000040
045400  025052  020040     000  AS2SP2: .ASCIZ  /**  /
045405     125  044516  052502  EM1:    .ASCIZ  /UNIBUS PARITY ERROR/
045412  020123  040520  044522
045420  054524  042440  051122
045426  051117     000
045431     116  047117  042455  EM2:    .ASCIZ  /NON-EXISTANT MEMORY/
045436  044530  052123  047101
045444  020124  042515  047515
045452  054522     000
045455     11?  047117  042455  EM3:    .ASCIZ  /NON-EXISTANT DRIVE/
045462  044530  052123  047101
045470  020124  051104  053111
045476  000105
```

```
 9014  045500  047125  052111  043040  EM4:    .ASCIZ  /UNIT FIELD ERROR/
 9015  045506  042511  042114  042440
 9016  045514  051122  051117     000
 9017  045521     123  041125  054523  EM5:    .ASCIZ  /SUBSYS TIMEOUT/
 9018  045526  020123  044524  042515
 9019  045534  052517  000124
 9020  045540  020104  047524  041440  EM6:    .ASCIZ  /D TO C PARITY ERROR/
 9021  045546  050040  051101  052111
 9022  045554  020131  051105  047522
 9023  045562  000122
 9024  045564  051104  053111  020105  EM7:    .ASCIZ  /DRIVE DETECTED PARITY ERROR/
 9025  045572  042504  042524  052103
 9026  045600  042105  050040  051101
 9027  045606  052111  020131  051105
 9028  045614  047522  000122
 9029  045620  041501  046040  053517  EM10:   .ASCIZ  /AC LOW/
 9030  045626     000
 9031  045627     123  042520  042105  EM11:   .ASCIZ  /SPEED LOSS/
 9032  045634  046040  051517  000123
 9033  045642  046111  042514  040507  EM12:   .ASCIZ  /ILLEGAL FUNCTION/
 9034  045650  020114  052506  041516
 9035  045656  044524  047117     000
 9036  045663     120  047522  051107  EM13:   .ASCIZ  /PROGRAMMING ERROR/
 9037  045670  046501  044515  043516
 9038  045676  042440  051122  051117
 9039  045704     000
 9040  045705     116  047117  042455  EM14:   .ASCIZ  /NON-EXISTANT FUNCTION/
 9041  045712  044530  052123  047101
 9042  045720  020124  052506  041516
 9043  045726  044524  047117     000
 9044  045733     104  044522  042526  EM15:   .ASCIZ  /DRIVE TYPE ERROR/
 9045  045740  052040  050131  020105
 9046  045746  051105  047522  000122
 9047  045754  047506  046522  052101  EM16:   .ASCIZ  /FORMAT ERROR/
 9048  045762  042440  051122  051117
 9049  045770     000
 9050  045771     127  044522  042524  EM17:   .ASCIZ  /WRITE LOCK ERROR/
 9051  045776  046040  041517  020113
 9052  046004  051105  047522  000122
 9053  046012  051104  053111  020105  EM20:   .ASCIZ  /DRIVE UNSAFE/
 9054  046020  047125  040523  042506
 9055  046026     000
 9056  046027     123  042505  020113  EM21:   .ASCIZ  /SEEK INCOMPLETE/
 9057  046034  047111  047503  050115
 9058  046042  042514  042524     000
 9059  046047     103  046131  047111  EM22:   .ASCIZ  /CYLINDER OVERFLOW/
 9060  046054  042504  020122  053117
 9061  046062  051105  046106  053517
 9062  046070     000
 9063  046071     111  046114  043505  EM23:   .ASCIZ  /ILLEGAL CYLINDER ADDRESS/
 9064  046076  046101  041440  046131
 9065  046104  047111  042504  020122
 9066  046112  042101  051104  051505
 9067  046120  000123
 9068  046122  051104  053111  020105  EM24:   .ASCIZ  /DRIVE OFF TRACK/
 9069  046130  043117  020106  051124
```

```
 9070  046136  041501  000113
 9071  046142  051104  053111  020105  EM25:   .ASCIZ  /DRIVE TIMING ERROR/
 9072  046150  044524  044515  043516
 9073  046156  042440  051122  051117
 9074  046164     000
 9075  046165     104  052101  020101  EM26:   .ASCIZ  /DATA LATE/
 9076  046172  040514  042524     000
 9077  046177     103  047117  051124  EM27:   .ASCIZ  /CONTROLLER TIMEOUT/
 9078  046204  046117  042514  020122
 9079  046212  044524  042515  052517
 9080  046220  000124
 9081  046222  050117  051105  052101  EM30:   .ASCIZ  /OPERATION INCOMPLETE/
 9082  046230  047511  020116  047111
 9083  046236  047503  050115  042514
 9084  046244  042524     000
 9085  046247     110  040505  042504  EM31:   .ASCIZ  /HEADER VRC ERROR/
 9086  046254  020122  051126  020103
 9087  046262  051105  047522  000122
 9088  046270  040504  040524  041440  EM32:   .ASCIZ  /DATA CHECK ERROR/
 9089  046276  042510  045503  042440
 9090  046304  051122  051117     000
 9091  046311     127  044522  042524  EM33:   .ASCIZ  /WRITE CHECK ERROR/
 9092  046316  041440  042510  045503
 9093  046324  042440  051122  051117
 9094  046332     000
 9095  046333     104  052101  020101  EM34:   .ASCIZ  /DATA MISCOMPARE/
 9096  046340  044515  041523  046517
 9097  046346  040520  042522     000
 9098  046353     116  020117  051104  EM35:   .ASCIZ  /NO DRIVE RESPONSE-UFE & NXD/
 9099  046360  053111  020105  042522
 9100  046366  050123  047117  042523
 9101  046374  052455  042506  023040
 9102  046402  047040  042130     000
 9103  046407     104  044522  042526  EM36:   .ASCIZ  /DRIVE ERROR WILL NOT CLEAR/
 9104  046414  042440  051122  051117
 9105  046422  053440  046111  020114
 9106  046430  047516  020124  046103
 9107  046436  040505  000122
 9108  046442  051104  053111  020105  EM37:   .ASCIZ  /DRIVE STATUS CHANGE WILL NOT CLEAR/
 9109  046450  052123  052101  051525
 9110  046456  041440  040510  043516
 9111  046464  020105  044527  046114
 9112  046472  047040  052117  041440
 9113  046500  042514  051101     000
 9114  046505     101  052124  047047  EM40:   .ASCIZ  /ATT'N BUT NO STATUS CHANGE OR FAULT/
 9115  046512  041040  052125  047040
 9116  046520  020117  052123  052101
 9117  046526  051525  041440  040510
 9118  046534  043516  020105  051117
 9119  046542  043040  052501  052114
 9120  046550     000
 9121  046551     101  052124  047047  EM41:   .ASCIZ  /ATT'N BUT DRIVE NOT AVAILABLE/
 9122  046556  041040  052125  042040
 9123  046564  044522  042526  047040
 9124  046572  052117  040440  040526
 9125  046600  046111  041101  042514
```

```
 9126  046606     000
 9127  046607     101  052124  047047  EM42:   .ASCIZ  /ATT'N WHEN NOT EXPECTED/
 9128  046614  053440  042510  020116
 9129  046622  047516  020124  054105
 9130  046630  042520  052103  042105
 9131  046636     000
 9132  046637     105  051122  051117  EM43:   .ASCIZ  /ERROR GATHERING DRIVE STATUS/
 9133  046644  043440  052101  042510
 9134  046652  044522  043516  042040
 9135  046660  044522  042526  051440
 9136  046666  040524  052524  000123
 9137  046674  052515  052114  050111  EM52:   .ASCIZ  /MULTIPLE DRIVE SELECT/
 9138  046702  042514  042040  044522
 9139  046710  042526  051440  046105
 9140  046716  041505  000124
 9141  046722  042510  042101  051105  EM53:   .ASCIZ  /HEADER COMPARE ERROR/
 9142  046730  041440  046517  040520
 9143  046736  042522  042440  051122
 9144  046744  051117     000
 9145  046747     123  041125  054523  EM56:   .ASCIZ  /SUBSYS TIMEOUT/
 9146  046754  020123  044524  042515
 9147  046762  052517  000124
 9148  046766  051105  047522  020122  EM60:   .ASCIZ  /ERROR IN RECAL FOR RECOVERY/
 9149  046774  047111  051040  041505
 9150  047002  046101  043040  051117
 9151  047010  051040  041505  053117
 9152  047016  051105  000131
 9153  047022  051120  043517  040522  EM61:   .ASCIZ  /PROGRAM ABORTING FATAL ERROR IN RETRY/
 9154  047030  020115  041101  051117
 9155  047036  044524  043516  043040
 9156  047044  052101  046101  042440
 9157  047052  051122  051117  044440
 9158  047060  020116  042522  051124
 9159  047066  000131
 9160  047070  054503  044514  042116  EM62:   .ASCIZ  /CYLINDER MISCOMPARE/
 9161  047076  051105  046440  051511
 9162  047104  047503  050115  051101
 9163  047112  000105
 9164  047114  046103  040505  020122  EM63:   .ASCIZ  /CLEAR CONTROLLER DID NOT CLEAR ERROR/
 9165  047122  047503  052116  047522
 9166  047130  046114  051105  042040
 9167  047136  042111  047040  052117
 9168  047144  041440  042514  051101
 9169  047152  042440  051122  051117
 9170  047160     000
 9171  047161     116  020117  052101  EM64:   .ASCIZ  /NO ATT'N IN ATT'N SUMMARY REGISTER/
 9172  047166  023524  020116  047111
 9173  047174  040440  052124  047047
 9174  047202  051440  046525  040515
 9175  047210  054522  051040  043505
 9176  047216  051511  042524  000122
 9177  047224  047125  047523  044514  EM65:   .ASCIZ  /UNSOLICITED ATTENTION/
 9178  047232  044503  042524  020104
 9179  047240  052101  042524  052116
 9180  047246  047511  000116
 9181  047252  047125  054105  042520  EM66:   .ASCIZ  /UNEXPECTED DATA TYPE ERROR/
```

```
 9182  047260  052103  042105  042040
 9183  047266  052101  020101  054524
 9184  047274  042520  042440  051122
 9185  047302  051117     000
 9186  047305     101  052124  047047  EM67:   .ASCIZ  /ATT'N DID NOT RESET WITH CLEAR/
 9187  047312  042040  042111  047040
 9188  047320  052117  051040  051505
 9189  047326  052105  053440  052111
 9190  047334  020110  046103  040505
 9191  047342  000122
 9192  047344  052523  051502  051531  EM70:   .ASCIZ  /SUBSYS CLEAR DID NOT CLEAR DRIVE ATT'N/
 9193  047352  041440  042514  051101
 9194  047360  042040  042111  047040
 9195  047366  052117  041440  042514
 9196  047374  051101  042040  044522
 9197  047402  042526  040440  052124
 9198  047410  047047     000
 9199  047413     104  052101  020101  EM71:   .ASCIZ  /DATA LATE WHEN UNLOADING HEADER/
 9200  047420  040514  042524  053440
 9201  047426  042510  020116  047125
 9202  047434  047514  042101  047111
 9203  047442  020107  042510  042101
 9204  047450  051105     000
 9205  047453     103  047117  051124  EM72:   .ASCIZ  /CONTROLLER ERROR WHEN DRIVER SERVICING/
 9206  047460  046117  042514  020122
 9207  047466  051105  047522  020122
 9208  047474  044127  047105  042040
 9209  047502  044522  042526  020122
 9210  047510  042523  053122  041511
 9211  047516  047111  000107
 9212  047522  051104  053111  020105  EM73:   .ASCIZ  /DRIVE DETECTED PARITY ERROR/
 9213  047530  042504  042524  052103
 9214  047536  042105  050040  051101
 9215  047544  052111  020131  051105
 9216  047552  047522  000122
 9217  047556  047125  042504  044506  EM74:   .ASCIZ  /UNDEFINED ERROR/
 9218  047564  042516  020104  051105
 9219  047572  047522  000122
 9220  047576  040515  045522  047111  EM75:   .ASCIZ  /MARKING THIS SECTOR BAD/
 9221  047604  020107  044124  051511
 9222  047612  051440  041505  047524
 9223  047620  020122  040502  000104
 9224  047626  040502  020104  040504  EM76:   .ASCIZ  /BAD DATA VERIF'N WITH READ. ECC OF LAST RETRY IS:/
 9225  047634  040524  053040  051105
 9226  047642  043111  047047  053440
 9227  047650  052111  020110  042522
 9228  047656  042101  020056  041505
 9229  047664  020103  043117  046040
 9230  047672  051501  020124  042522
 9231  047700  051124  020131  051511
 9232  047706  000072
 9233  047710  042522  051124  020131  EM77:   .ASCIZ  /RETRY SUCCESSFUL/
 9234  047716  052523  041503  051505
 9235  047724  043123  046125     000
 9236  047731     122  052105  054522  EM100:  .ASCIZ  /RETRY UNSUCCESSFUL/
 9237  047736  052440  051516  041525
```

```
 9238  047744  042503  051523  052506
 9239  047752  000114
 9240  047754  040503  047116  052117  EM101:  .ASCIZ  /CANNOT FIND A VALID HEADER IN TRACK JUST READ/
 9241  047762  043040  047111  020104
 9242  047770  020101  040526  044514
 9243  047776  020104  042510  042101
 9244  050004  051105  044440  020116
 9245  050012  051124  041501  020113
 9246  050020  052512  052123  051040
 9247  050026  040505  000104
 9248  050032  040502  020104  042523  EM102:  .ASCIZ  /BAD SECTOR ERROR ON SECTOR NOT LISTED BAD/
 9249  050040  052103  051117  042440
 9250  050046  051122  051117  047440
 9251  050054  020116  042523  052103
 9252  050062  051117  047040  052117
 9253  050070  046040  051511  042524
 9254  050076  020104  040502  000104
 9255  050104  044524  042515  026504  EM103:  .ASCIZ  /TIMED-OUT ON READ HDR/
 9256  050112  052517  020124  047117
 9257  050120  051040  040505  020104
 9258  050126  042110  000122
 9259  050132  044524  042515  026504  EM104:  .ASCIZ  /TIMED-OUT ON SEEK/
 9260  050140  052517  020124  047117
 9261  050146  051440  042505  000113
 9262  050154  051104  053111  020105  EM105:  .ASCIZ  /DRIVE SIEZED BY OTHER PORT/
 9263  050162  044523  055105  042105
 9264  050170  041040  020131  052117
 9265  050176  042510  020122  047520
 9266  050204  052122     000
 9267  050207     104  052101  020101  EM106:  .ASCIZ  /DATA MISCMPR WHILE BAI SET/
 9268  050214  044515  041523  050115
 9269  050222  020122  044127  046111
 9270  050230  020105  040502  020111
 9271  050236  042523  000124
 9272  050242  047516  047040  046505  EM107:  .ASCIZ  /NO NEM ERROR WHEN REF'ING LOC 760000/
 9273  050250  042440  051122  051117
 9274  050256  053440  042510  020116
 9275  050264  042522  023506  047111
 9276  050272  020107  047514  020103
 9277  050300  033067  030060  030060
 9278  050306     000
 9279  050307     111  052116  050122  EM110:  .ASCIZ  /INTRPT WHEN CNTRLR NOT RDY/
 9280  050314  020124  044127  047105
 9281  050322  041440  052116  046122
 9282  050330  020122  047516  020124
 9283  050336  042122  000131
 9284  050342  047516  040440  052124  EM111:  .ASCIZ  /NO ATT'N ON SEEK/
 9285  050350  047047  047440  020116
 9286  050356  042523  045505     000
 9287  050363     104  044522  042526  EM112:  .ASCIZ  /DRIVE'S CYLINDER INCORRECT/
 9288  050370  051447  041440  046131
 9289  050376  047111  042504  020122
 9290  050404  047111  047503  051122
 9291  050412  041505  000124
 9292  050416  041101  051117  026524  EM113:  .ASCIZ  /ABORT- CAN'T READ BSF/
 9293  050424  041440  047101  052047
```

```
 9294  050432  051040  040505  020104
 9295  050440  051502  000106
 9296  050444  052113  030461  043040  EM114:  .ASCIZ  /KT11 FAILURE/
 9297  050452  044501  052514  042522
 9298  050460     000
 9299  050461     115  046505  050040  EM115:  .ASCIZ  /MEM PARITY ERR/
 9300  050466  051101  052111  020131
 9301  050474  051105  000122
 9302  050500  052503  051122  047105  DH100:  .ASCIZ  /CURRENT COMMAND :/
 9303  050506  020124  047503  046515
 9304  050514  047101  020104  000072
 9305  050522  051120  053105  047511  DH105:  .ASCIZ  /PREVIOUS COMMAND :/
 9306  050530  051525  041440  046517
 9307  050536  04C515  042116  035040
 9308  050544     000
 9309  050545     122  033113  030461  DH200:  .ASCIZ  /RK611 REGISTERS :/
 9310  050552  051040  043505  051511
 9311  050560  042524  051522  035040
 9312  050566     000
 9313  050567     122  046505  044501  DH500:  .ASCIZ  /REMAINING REGISTERS NOT VALID/
 9314  050574  044516  043516  051040
 9315  050602  043505  051511  042524
 9316  050610  051522  047040  052117
 9317  050616  053040  046101  042111
 9318  050624     000
 9319  050625     124  042510  043040  DH501:  .ASCIZ  /THE FOLLOWING REGISTER DATA MAY BE INCORRECT :/
 9320  050632  046117  047514  044527
 9321  050640  043516  051040  043505
 9322  050646  051511  042524  020122
 9323  050654  040504  040524  046440
 9324  050662  054501  041040  020105
 9325  050670  047111  047503  051122
 9326  050676  041505  020124  000072
 9327  050704  051105  020122  041520  DH101:  .ASCIZ  /ERR PC  DRIVE   CMND    CYLNDR  TRACK   SECTOR  WD CNT/
 9328  050712  020040  051104  053111
 9329  050720  020105  020040  046503
 9330  050726  042116  020040  020040
 9331  050734  054503  047114  051104
 9332  050742  020040  051124  041501
 9333  050750  020113  020040  042523
 9334  050756  052103  051117  020040
 9335  050764  042127  041440  052116
 9336  050772     000
 9337  050773     110  020111  040502  DH102:  .ASCIZ  /HI BA   LO BA/
 9338  051000  020040  046040  020117
 9339  051006  040502     000
 9340  051011     120  042522  027126  DH103:  .ASCIZ  /PREV. UNIBUS MAP :/
 9341  051016  052440  044516  052502
 9342  051024  020123  040515  020120
 9343  051032  000072
 9344  051034  044510  051040  043505  DH104:  .ASCIZ  /HI REGO LO REGO/
 9345  051042  020060  047514  051040
 9346  051050  043505  000060
 9347  051054  052503  027122  052440  DH106:  .ASCIZ  /CUR. UNIBUS MAP :/
 9348  051062  044516  052502  020123
 9349  051070  040515  020120  000072
```

```
 9350  051076  045522  051503  020061  DH201:  .ASCIZ  /RKCS1   RKCS2   RKDC    RKDA    RKWCR   RKBA    RKASOF/
 9351  051104  020040  045522  051503
 9352  051112  020062  020040  045522
 9353  051120  041504  020040  020040
 9354  051126  045522  040504  020040
 9355  051134  020040  045522  041527
 9356  051142  020122  020040  045522
 9357  051150  040502  020040  020040
 9358  051156  045522  051501  043117
 9359  051164     000
 9360  051165     122  042113  020123  DH202:  .ASCIZ  /RKDS    RKER/
 9361  051172  020040  051040  042513
 9362  051200  000122
 9363  051202  030101  020060  020040  DH203:  .ASCIZ  /A00     B00     A01     B01     A02     B02     A03     B03/
 9364  051210  020040  030102  020060
 9365  051216  020040  020040  030101
 9366  051224  020061  020040  020040
 9367  051232  030102  020061  020040
 9368  051240  020040  030101  020062
 9369  051246  020040  020040  030102
 9370  051254  020062  020040  020040
 9371  051262  030101  020063  020040
 9372  051270  020040  030102  000063
 9373  051276  042510  042101  051105  DH601:  .ASCIZ  /HEADER SHOULD BE:/
 9374  051304  051440  047510  046125
 9375  051312  020104  042502  000072
 9376  051320  042510  042101  051105  DH602:  .ASCIZ  /HEADER OF SECTOR 0 THIS CYLINDER IS:/
 9377  051326  047440  020106  042523
 9378  051334  052103  051117  030040
 9379  051342  052040  044510  020123
 9380  051350  054503  044514  042116
 9381  051356  051105  044440  035123
 9382  051364     000
 9383  051365     120  041501  020113  DH604:  .ASCIZ  /PACK ADDRESS OF ERROR(S) :/
 9384  051372  042101  051104  051505
 9385  051400  020123  043117  042440
 9386  051406  051122  051117  051450
 9387  051414  020051  000072
 9388  051420  054503  047114  051104  DH5041: .ASCIZ  /CYLNDR  TRACK   SECTOR/
 9389  051426  020040  051124  041501
 9390  051434  020113  020040  042523
 9391  051442  052103  051117     000
 9392  051447     103  046131  042116  DH6042: .ASCIZ  /CYLNDR  TRACK/
 9393  051454  020122  052040  040522
 9394  051462  045503     000
 9395  051465     105  041503  042040  DH605:  .ASCIZ  /ECC DATA IS:/
 9396  051472  052101  020101  051511
 9397  051500  000072
 9398  051502  047520  020123  020040  DH6051: .ASCIZ  /POS     PAT     CORRECTABLE?/
 9399  051510  020040  040520  020124
 9400  051516  020040  020040  047503
 9401  051524  051122  041505  040524
 9402  051532  046102  037505     000
 9403  051537     127  051117  030504  DH606:  .ASCIZ  /WORD1   WORD2   WORD3/
 9404  051544  020040  053440  051117
 9405  051552  031104  020040  053440
```

```
 9406  051560  051117  031504     000
 9407  051565     102  042101  044040  DH607:  .ASCIZ  /BAD HEADER IS :/
 9408  051572  040505  042504  020122
 9409  051600  051511  035040     000
 9410  051605     127  020104  020043  DH701:  .ASCIZ  /WD #    GOOD    BAD     HI PHY  LO PHY  VRT AD  KIPAR6/
 9411  051612  020040  043440  047517
 9412  051620  020104  020040  041040
 9413  051626  042101  020040  020040
 9414  051634  044040  020111  044120
 9415  051642  020131  046040  020117
 9416  051650  044120  020131  053040
 9417  051656  052122  040440  020104
 9418  051664  045440  050111  051101
 9419  051672  000066
 9420  051674  047507  042117  020040  DH702:  .ASCIZ  /GOOD    BAD/
 9421  051702  020040  040502  000104
 9422  051710  040506  046111  047111  DH703:  .ASCIZ  /FAILING DATA WORD :/
 9423  051716  020107  040504  040524
 9424  051724  053440  051117  020104
 9425  051732  000072
 9426  051734  051123  004460  051123  DH704:  .ASCIZ  /SR0    SR1     SR2     SR3/
 9427  051742  004461  051123  004462
 9428  051750  051123  000063
 9429  051754  052113  030461  051040  DH705:  .ASCIZ  /KT11 REGS :/
 9430  051762  043505  020123  000072
 9431  051770  042515  020115  042522  DH706:  .ASCIZ  /MEM REGS :/
 9432  051776  051507  035040     000
 9433  052003     120  004503  047514  DH707:  .ASCIZ  /PC     LOERAD  HIERAD  MEMSYS/
 9434  052010  051105  042101  044011
 9435  052016  042511  040522  004504
 9436  052024  042515  051515  051531
 9437  052032     000
 9438  052033     120  004503  051503  DH710:  .ASCIZ  /PC     CSR AD  CSR/
 9439  052040  020122  042101  041411
 9440  052046  051123     000
 9441  052051     103  046131  047111  DH711:  .ASCIZ  /CYLINDERS :/
 9442  052056  042504  051522  035040
 9443  052064     000
 9444  052065     122  042113  020123  DH204:  .ASCIZ  /RKDS    RKER    RKMR2   RKMR3/
 9445  052072  020040  051040  042513
 9446  052100  020122  020040  051040
 9447  052106  046513  031122  020040
 9448  052114  051040  046513  031522
 9449  052122     000
 9450  052123     105  051122  051117  DH502:  .ASCIZ  /ERROR TRYING TO GET FAILURE INFO/
 9451  052130  052040  054522  047111
 9452  052136  020107  047524  043440
 9453  052144  052105  043040  044501
 9454  052152  052514  042522  044440
 9455  052160  043116  000117
 9456  052164  042523  047503  042116  DH503:  .ASCIZ  /SECOND TIMEOUT OCCURRED GETTING DRIVE STATUS/
 9457  052172  052040  046511  047505
 9458  052200  052125  047440  041503
 9459  052206  051125  042522  020104
 9460  052214  042507  052124  047111
 9461  052222  020107  051104  053111
```

```
 9462  052230  020105  052123  052101
 9463  052236  051525     000
 9464  052241     116  046525  042502  DH800:  .ASCIZ  /NUMBER OF RETRIES:/
 9465  052246  020122  043117  051040
 9466  052254  052105  044522  051505
 9467  052262  000072
 9468                                  .EVEN
 9469  052264  001116  005430  005426  DT100:  .WORD   $ERRPC,DRIVE,ERRCOM,$REG0,$REG1,$REG2,$REG3
 9470  052272  001162  001164  001166
 9471  052300  001170
 9472  052302  001256  001260          DT102:  .WORD   $REG36,$REG37
 9473  052306  001174  001176  001200  DT201:  .WORD   $REG5,$REG6,$REG7,$REG10,$REG11,$REG12,$REG13
 9474  052314  001202  001204  001206
 9475  052322  001210
 9476  052324  001212  001214          DT202:  .WORD   $REG14,$REG15
 9477  052330  001216  001220  001222  DT203:  .WORD   $REG16,$REG17,$REG20,$REG21,$REG22,$REG23,$REG24,$REG25
 9478  052336  001224  001226  001230
 9479  052344  001232  001234
 9480  052350  001174  001176  001200  DT601:  .WORD   $REG5,$REG6,$REG7
 9481  052356  001202  001204  001206  DT602:  .WORD   $REG10,$REG11,$REG12,$REG13,$REG14,$REG15,$REG16
 9482  052364  001210  001212  001214
 9483  052372  001216
 9484
 9485  052374  000006                  DF01:   .WORD   6               ;NUMBER OF HEADER LINES
 9486  052376     000                          .BYTE   0               ;NUMBER OF COL FOR FIRST HDR
 9487  052377     000                          .BYTE   0               ;ALL CHARACTERS OCTAL
 9488  052400  050704                          .WORD   DH101           ;SECOND HDR LINE
 9489  052402     007     000                  .BYTE   7,0             ;NUM OF COL-ALL OCTAI
 9490  052404  050773                          .WORD   DH102
 9491  052406     002     000                  .BYTE   2,0
 9492  052410  050545                          .WORD   DH200
 9493  052412     000     000                  .BYTE   0,0
 9494  052414  051076                          .WORD   DH201
 9495  052416     007     000                  .BYTE   7,0
 9496  052420  050567                          .WORD   DH500
 9497  052422     000     000                  .BYTE   0,0
 9498
 9499  052424  000007                  DF02:   .WORD   7
 9500  052426     000     000                  .BYTE   0,0
 9501  052430  050704                          .WORD   DH101
 9502  052432     007     000                  .BYTE   7,0
 9503  052434  050773                          .WORD   DH102
 9504  052436     002     000                  .BYTE   2,0
 9505  052440  050545                          .WORD   DH200
 9506  052442     000     000                  .BYTE   0,0
 9507  052444  051076                          .WORD   DH201
 9508  052446     007     000                  .BYTE   7,0
 9509  052450  051165                          .WORD   DH202
 9510  052452     002     000                  .BYTE   2,0
 9511  052454  051202                          .WORD   DH203
 9512  052456     010     000                  .BYTE   10,0
 9513
 9514  052460  000010                  DF03:   .WORD   10
 9515  052462     000     000                  .BYTE   0,0
 9516  052464  050704                          .WORD   DH101
 9517  052466     007     000                  .BYTE   7,0
```

```
 9518  052470  050773                          .WORD   DH102
 9519  052472     002     000                  .BYTE   2,0
 9520  052474  050545                          .WORD   DH200
 9521  052476     000     000                  .BYTE   0,0
 9522  052500  051076                          .WORD   DH201
 9523  052502     007     000                  .BYTE   7,0
 9524  052504  050625                          .WORD   DH501           ;''THE FOLLOWING REG DATA MB INCORRECT''
 9525  052506     000     000                  .BYTE   0,0
 9526  052510  051165                          .WORD   DH202
 9527  052512     002     000                  .BYTE   2,0
 9528  052514  051202                          .WORD   DH203
 9529  052516     010     000                  .BYTE   10,0
 9530
 9531  052520  000003                  DF04:   .WORD   3
 9532  052522     000     000                  .BYTE   0,0
 9533  052524  050704                          .WORD   DH101
 9534  052526     007     000                  .BYTE   7,0
 9535  052530  050773                          .WORD   DH102
 9536  052532     002     000                  .BYTE   2,0
 9537
 9538  052534  000007                  DF05:   .WORD   7               ;OPI, HVRC
 9539  052536     000     000                  .BYTE   0,0
 9540  052540  050704                          .WORD   DH101
 9541  052542     007     000                  .BYTE   7,0
 9542  052544  050773                          .WORD   DH102
 9543  052546     002     000                  .BYTE   2,0
 9544  052550  051276                          .WORD   DH601
 9545  052552     000     000                  .BYTE   0,0
 9546  052554  051537                          .WORD   DH606
 9547  052556     003     000                  .BYTE   3,0
 9548  052560  051320                          .WORD   DH602
 9549  052562     000     000                  .BYTE   0,0
 9550  052564  051537                          .WORD   DH606
 9551  052566     003     000                  .BYTE   3,0
 9552
 9553  052570  000007                  DF07:   .WORD   7
 9554  052572     000     000                  .BYTE   0,0
 9555  052574  050704                          .WORD   DH101
 9556  052576     007     000                  .BYTE   7,0
 9557  052600  050773                          .WORD   DH102
 9558  052602     002     000                  .BYTE   2,0
 9559  052604  051365                          .WORD   DH604
 9560  052606     000     000                  .BYTE   0,0
 9561  052610  051420                          .WORD   DH6041
 9562  052612     003     000                  .BYTE   3,0
 9563  052614  051465                          .WORD   DH605
 9564  052616     000     000                  .BYTE   0,0
 9565  052620  051502                          .WORD   DH6051
 9566  052622     003     000                  .BYTE   3,0
 9567
 9568  052624  000005                  DF10:   .WORD   5
 9569  052626     000     000                  .BYTE   0,0
 9570  052630  050704                          .WORD   DH101
 9571  052632     007     000                  .BYTE   7,0
 9572  052634  050773                          .WORD   DH102
 9573  052636     002     000                  .BYTE   2,0
```

```
 9574  052640  051365                        .WORD   DH604
 9575  052642     000      000               .BYTE   0,0
 9576  052644  051420                        .WORD   DH6041
 9577  052646     003      000               .BYTE   3,0
 9578
 9579  052650  000004                  DF11: .WORD   4
 9580  052652     000      000               .BYTE   0,0
 9581  052654  051365                        .WORD   DH604
 9582  052656     000      000               .BYTE   0,0
 9583  052660  051420                        .WORD   DH6041
 9584  052662     003      000               .BYTE   3,0
 9585  052664  051605                        .WORD   DH701
 9586  052666     000      000               .BYTE   0,0
 9587
 9588  052670  000006                  DF12: .WORD   6
 9589  052672     000      000               .BYTE   0,0
 9590  052674  050704                        .WORD   DH101
 9591  052676     007      000               .BYTE   7,0
 9592  052700  050773                        .WORD   DH102
 9593  052702     002      000               .BYTE   2,0
 9594  052704  050545                        .WORD   DH200
 9595  052706     000      000               .BYTE   0,0
 9596  052710  051076                        .WORD   DH201
 9597  052712     007      000               .BYTE   7,0
 9598  052714  052065                        .WORD   DH204
 9599  052716     004      000               .BYTE   4,0
 9600
 9601  052720  000006                  DF13: .WORD   6               ;FORMAT FOR 2ND LEVEL ERROR
 9602  052722     000      000               .BYTE   0,0             ;IN HEADER COMPARE ERROR
 9603  052724  050704                        .WORD   DH101           ;AND 2ND LEVEL HEADER
 9604  052726     007      000               .BYTE   7,0             ;VRC ERROR
 9605  052730  050773                        .WORD   DH102
 9606  052732     002      000               .BYTE   2,0
 9607  052734  051276                        .WORD   DH601
 9608  052736     000      000               .BYTE   0,0
 9609  052740  051537                        .WORD   DH606
 9610  052742     003      000               .BYTE   3,0
 9611  052744  052123                        .WORD   DH502
 9612  052746     000      000               .BYTE   0,0
 9613
 9614  052750  000006                  DF14: .WORD   6               ;FORMAT FOR 2ND LEVEL ERROR
 9615  052752     000      000               .BYTE   0,0             ;IN OPERATION INCOMPLETE ERROR
 9616  052754  050704                        .WORD   DH101
 9617  052756     007      000               .BYTE   7,0
 9618  052760  050773                        .WORD   DH102
 9619  052762     002      000               .BYTE   2,0
 9620  052764  051276                        .WORD   DH601
 9621  052766     000      000               .BYTE   0,0
 9622  052770  051537                        .WORD   DH606
 9623  052772     003      000               .BYTE   3,0
 9624  052774  052123                        .WORD   DH502
 9625  052776     000      000               .BYTE   0,0
 9626
 9627  053000  000011                  DF15: .WORD   11
 9628  053002     000      000               .BYTE   0,0
 9629  053004  050704                        .WORD   DH101
```

```
 9630  053006     007     000                  .BYTE   7,0
 9631  053010  050773                          .WORD   DH102
 9632  053012     002     000                  .BYTE   2,0
 9633  053014  050545                          .WORD   DH200
 9634  053016     000     000                  .BYTE   0,0
 9635  053020  051076                          .WORD   DH201
 9636  053022     007     000                  .BYTE   7,0
 9637  053024  051165                          .WORD   DH202
 9638  053026     002     000                  .BYTE   2,0
 9639  053030  052164                          .WORD   DH503
 9640  053032     000     000                  .BYTE   0,0
 9641  053034  050625                          .WORD   DH501
 9642  053036     000     000                  .BYTE   0,0
 9643  053040  051202                          .WORD   DH203
 9644  053042     010     000                  .BYTE   10,0
 9645
 9646  053044  000011                  DF16:   .WORD   11
 9647  053046     000     000                  .BYTE   0,0
 9648  053050  050704                          .WORD   DH101
 9649  053052     007     000                  .BYTE   7,0
 9650  053054  050773                          .WORD   DH102
 9651  053056     002     000                  .BYTE   2,0
 9652  053060  050545                          .WORD   DH200
 9653  053062     000     000                  .BYTE   0,0
 9654  053064  051076                          .WORD   DH201
 9655  053066     007     000                  .BYTE   7,0
 9656  053070  051165                          .WORD   DH202
 9657  053072     002     000                  .BYTE   2,0
 9658  053074  052123                          .WORD   DH502
 9659  053076     000     000                  .BYTE   0,0
 9660  053100  050625                          .WORD   DH501
 9661  053102     000     000                  .GYTE   0,0
 9662  053104  051202                          .WORD   DH203
 9663  053106     010     000                  .BYTE   10,0
 9664
 9665  053110  000005                  DF17:   .WORD   5
 9666  053112     000     000                  .BYTE   0,0
 9667  053114  050704                          .WORD   DH101
 9668  053116     007     000                  .BYTE   7,0
 9669  053120  050773                          .WORD   DH102
 9670  053122     002     000                  .BYTE   2,0
 9671  053124  052051                          .WORD   DH711
 9672  053126     000     000                  .BYTE   0,0
 9673  053130  051674                          .WORD   DH702
 9674  053132     002     000                  .BYTE   2,0
 9675
 9676  053134  000002                  DF20:   .WORD   2
 9677  053136     000     000                  .BYTE   0,0
 9678  053140  051034                          .WORD   DH104
 9679  053142     002     000                  .BYTE   2,0
 9680
 9681  053144  000002                  DF21:   .WORD   2
 9682  053146     000     000                  .BYTE   0,0
 9683  053150  051502                          .WORD   DH6051
 9684  053152     003     000                  .BYTE   3,0
 9685
```

```
 9686  053154  000006                  DF22:   .WORD   6
 9687  053156     000      000                 .BYTE   0,0
 9688  053160  050500                          .WORD   DH100
 9689  053162     000      000                 .BYTE   0,0
 9690  053164  050704                          .WORD   DH101
 9691  053166     007      000                 .BYTE   7,0
 9692  053170  050773                          .WORD   DH102
 9693  053172     002      000                 .BYTE   2,0
 9694  053174  051054                          .WORD   DH106
 9695  053176     000      000                 .BYTE   0,0
 9696  053200  051034                          .WORD   DH104
 9697  053202     002      000                 .BYTE   2,0
 9698
 9699  053204  000002                  DF23:   .WORD   2
 9700  053206     000      000                 .BYTE   0,0
 9701  053210  052241                          .WORD   DH800
 9702  053212     001      000                 .BYTE   1,0
 9703
 9704  053214  000001                  DF24:   .WORD   1
 9705  053216     002      000                 .BYTE   2,0
 9706
 9707  053220  000000                  DF25:   .WORD   0
 9708  053222     007      000                 .BYTE   7,0
 9709
 9710  053224  000002                  DF26:   .WORD   2
 9711  053226     002      000                 .BYTE   2,0
 9712  053230  051202                          .WORD   DH203
 9713  053232     010      000                 .BYTE   10,0
 9714
 9715  053234  000005                  DF27:   .WORD   5
 9716  053236     000      000                 .BYTE   0,0
 9717  053240  050704                          .WORD   DH101
 9718  053242     007      000                 .BYTE   7,0
 9719  053244  050773                          .WORD   DH102
 9720  053246     002      000                 .BYTE   2,0
 9721  053250  051710                          .WORD   DH703
 9722  053252     000      000                 .BYTE   0,0
 9723  053254  051674                          .WORD   DH702
 9724  053256     002      000                 .BYTE   2,0
 9725
 9726  053260  000005                  DF30:   .WORD   5
 9727  053262     000      000                 .BYTE   0,0
 9728  053264  050704                          .WORD   DH101
 9729  053266     007      000                 .BYTE   7,0
 9730  053270  050773                          .WORD   DH102
 9731  053272     002      000                 .BYTE   2,0
 9732  053274  051754                          .WORD   DH705
 9733  053276     000      000                 .BYTE   0,0
 9734  053300  051734                          .WORD   DH704
 9735  053302     004      000                 .BYTE   4,0
 9736
 9737  053304  000005                  DF31:   .WORD   5
 9738  053306     000      000                 .BYTE   0,0
 9739  053310  050704                          .WORD   DH101
 9740  053312     007      000                 .BYTE   7,0
 9741  053314  050773                          .WORD   DH102
```

```
 9742  053316     002     000                  .BYTE   2,0
 9743  053320  051770                          .WORD   DH706
 9744  053322     000     000                  .BYTE   0,0
 9745  053324  052033                          .WORD   DH710
 9746  053326     003     000                  .BYTE   3,0
 9747
 9748  053330  000400                  RWBUF:  .BLKW   256.            ;READ/WRITE DATA BUFFER
 9749
 9750  054330  005015  020040  020040  NOTMSG: .ASCII  <15><12>/          *** NOTE ***/<15><12><12>
 9751  054336  020040  020040  020040
 9752  054344  025052  020052  047516
 9753  054352  042524  025040  025052
 9754  054360  005015     012
 9755  054363     117  046116  020131          .ASCII  /ONLY ALL RK06 OR ALL RK07'S MUST BE UNDER TEST/<15><12><12>
 9756  054370  046101  020114  045522
 9757  054376  033060  047440  020122
 9758  054404  046101  020114  045522
 9759  054412  033460  051447  046440
 9760  054420  051525  020124  042502
 9761  054426  052440  042116  051105
 9762  054434  052040  051505  006524
 9763  054442  005012
 9764  054444  046101  020114  051104          .ASCII  /ALL DRIVES TO BE TESTED MUST HAVE :/<15><12><12>
 9765  054452  053111  051505  052040
 9766  054460  020117  042502  052040
 9767  054466  051505  042524  020104
 9768  054474  052515  052123  044040
 9769  054502  053101  020105  006472
 9770  054510  005012
 9771  054512  027061  042040  051505          .ASCII  /1. DESIRED PORT SELECTED/<15><12>
 9772  054520  051111  042105  050040
 9773  054526  051117  020124  042523
 9774  054534  042514  052103  042105
 9775  054542  005015
 9776  054544  027062  053440  044522          .ASCII  /2. WRITE LOCK DISABLED/<15><12>
 9777  054552  042524  046040  041517
 9778  054560  020113  044504  040523
 9779  054566  046102  042105  005015
 9780  054574  051104  053111  051505          .ASCII  /DRIVES NOT TO BE TESTED MUST HAVE BOTH/<15><12>
 9781  054602  047040  052117  052040
 9782  054610  020117  042502  052040
 9783  054616  051505  042524  020104
 9784  054624  052515  052123  044040
 9785  054632  053101  020105  047502
 9786  054640  044124  005015
 9787  054644  047520  052122  020123          .ASCII  /PORTS DE-SELECTED./<15><12>
 9788  054652  042504  051455  046105
 9789  054660  041505  042524  027104
 9790  054666  005015
 9791  054670  044124  020105  040503          .ASCII  /THE CARTRIDGE MUST BE FORMATTED ON A KNOWN GOOD/<15><12>
 9792  054676  052122  044522  043504
 9793  054704  020105  052515  052123
 9794  054712  041040  020105  047506
 9795  054720  046522  052101  042524
 9796  054726  020104  047117  040440
 9797  054734  045440  047516  047127
```

```
 9798  054742  043440  047517  006504
 9799  054750     012
 9800  054751     127  046105  026514          .ASCII  /WELL-ALIGNED DRIVE, PRIOR TO TESTING./<15><12>
 9801  054756  046101  043511  042516
 9802  054764  020104  051104  053111
 9803  054772  026105  050040  044522
 9804  055000  051117  052040  020117
 9805  055006  042524  052123  047111
 9806  055014  027107  005015
 9807  055020  044124  020105  047503          .ASCII  /THE COMPLETE RUN TIME IS 8-10 MIN. PER DRIVE./<15><12>
 9808  055026  050115  042514  042524
 9809  055034  051040  047125  052040
 9810  055042  046511  020105  051511
 9811  055050  034040  030455  020060
 9812  055056  044515  027116  050040
 9813  055064  051105  042040  044522
 9814  055072  042526  006456     012
 9815  055077     124  020117  041101          .ASCIZ  /TO ABORT TESTING, TYPE CONTROL-C (^C)./<15><12><12>
 9816  055104  051117  020124  042524
 9817  055112  052123  047111  026107
 9818  055120  052040  050131  020105
 9819  055126  047503  052116  047522
 9820  055134  026514  020103  057050
 9821  055142  024503  006456  005012
 9822  055150     000
 9823
 9824
 9825
 9826
 9827
 9828
 9829            000001                  .END
```

```
ABASE = 177440           1496#   1592    1633    3595
ABORT = 000200           3243#   3833    4935    6635    6640    6723    6748    6845    6855    6860    6865    6870    6881
                         6887    6902    6937
ACDW1 = 000000           1592
ACDW2 = 000000           1592
ACLO  = 000010           2561#   6685
ACPUOP= 000000           1592    1607
ADDW0 = 000000           1592
ADDW1 = 000000           1592
ADDW10= 000000           1592
ADDW11= 000000           1592
ADDW12= 000000           1592
ADDW13= 000000           1592
ADDW14= 000000           1592
ADDW15= 000000           1592
ADDW2 = 000000           1592
ADDW3 = 000000           1592
ADDW4 = 000000           1592
ADDW5 = 000000           1592
ADDW6 = 000000           1592
ADDW7 = 000000           1592
ADDW8 = 000000           1592
ADDW9 = 000000           1592
ADEVCT= 000000           1592    1598
ADEVM = 000377           1497#   1592    1634
ADSCOR  017614           4111    4181    4243    4530#
AENV  = 000000           1592    1603
AENVM = 000000           1592    1604
AFATAL= 000000           1592    1595
ALLSYS  012752           3589    3612#   3625    3626    3669    3699    3700    3717    5054    5126    5793
ALLSY1  013010           3622#   3706
ALLSY2  013172           3662#   3672
ALLSY3  013170           3661#   3809
ALLTRM  033324           3834    4936    6724    6868    6875#
ALNPAK  011336           3415#   4995
AMADR1= 000000           1592    1620
AMADR2= 000000           1592    1624
AMADR3= 000000           1592    1627
AMADR4= 000000           1592    1630
AMAMS1= 000000           1592    1614
AMAMS2= 000000           1592    1622
AMAMS3= 000000           1592    1625
AMAMS4= 000000           1592    1628
AMSGAD= 000000           1592    1600
AMSGLG= 000000           1592    1601
AMSGTY= 000000           1592    1594
AMTYP1= 000000           1592    1615
AMTYP2= 000000           1592    1623
AMTYP3= 000000           1592    1626
AMTYP4= 000000           1592    1629
ANOTHR  010144           3295#   3697
ANYDER= 100000           3249#   4987    6354    6925
APASS = 000000           1592    1597
APRIOR= 000000           1592
APTCSU= 000040           8265    8910#
APTENV= 000001           8258    8704    8866    8908#
```

```
APTSIZ= 000200           3554    8907#
APTSPO= 000100           8260    8868    8909#
ASKSYS  013416           3692    3703    3708    3714#   3718    3719    3721    3726    3737    5058    5130    5807
ASKTYP  007775           3276#   5478
ASWREG= 000000           1592    1605
AS2SP2  045400           6393    9001#
ATESTN= 000000           1592    1596
AUNIT = 000000           1592    1599
AUSWR = 000000           1592    1606
AVECT1= 120210           1495#   1592    1631    3594
AVECT2= 000000           1592    1632
AVSCOR  017774           4119    4188    4252    4572#
A.ABNL  003056           2842#   3837*   3840*   4864*   4939*   4944*   5798*   5980*   5997*   6006*   6190*   6205*   6611*
                         6876*   6879*   6893*   6896*   6899*   7826
A.CONT  003060           2843#   7836
A.NORM  003054           2841#   4865*   6612*   6880*   7830
BADDRV  011253           3404#   4376*   4377*   4381    4942*   4947*   4948*   4949    4961*   4962*   4963    4969*   4970*
                         4971    4992*   4993*   4994
BADSEC= 001000           3245#   5722    5907    6270    6278    6818    6930
BAD632  012006           3474#   4989
BA1   = 000020           2520#   7995
BA16  = 000400           2496#
BA17  = 001000           2497#
BDSCHD  026762           5980    5997    6011#
BDSECT  011757           3468#   6029
BDSRCK  030020           5720    5905    6260#   6817
BGNRTY  033546           6535    6890    6901    6923#
BIT0  = 000001           1427#   2493    2510    2536    2558    2706    2850    2943    3789    5642    5691    5731    5918
                         6490    8110
BIT00 = 000001           1417#   1427
BIT01 = 000002           1416#   1426
BIT02 = 000004           1415#   1425
BIT03 = 000010           1414#   1424
BIT04 = 000020           1413#   1423
BIT05 = 000040           1412#   1422
BIT06 = 000100           1411#   1421
BIT07 = 000200           1410#   1420
BIT08 = 000400           1409#   1419    5533
BIT09 = 001000           1408#   1418    8714    8840
BIT1  = 000002           1426#   2511    2538    2707    2851    2944    3237    6494
BIT10 = 002000           1407#   2498    2527    2549    2584    2598    2611    2626    2640    2719    2860    3246    6525
                         8692
BIT11 = 004000           1406#   2499    2528    2550    2569    2585    2599    2612    2627    2641    2720    3247
BIT12 = 010000           1405#   2501    2529    2551    2586    2600    2613    2614    2628    2642    2721    3248    6405
                         6433
BIT13 = 020000           1404#   2502    2530    2552    2570    2587    2601    2615    2629    2643    2722    8699
BIT14 = 040000           1403#   2503    2531    2553    2571    2588    2602    2616    2630    2644    2723    2861    8824
BIT15 = 100000           1402#   2504    2505    2532    2554    2572    2589    2649    2724    2862    3249    6529
BIT2  = 000004           1425#   2512    2539    2540    2560    2708    2852    2945    3238    6498
BIT3  = 000010           1424#   2518    2519    2541    2561    2709    2853    2946    3239    6505
BIT4  = 000020           1423#   2513    2520    2542    2562    2563    2578    2620    2634    2710    2854    2947    3240
                         6509
BIT5  = 000040           1422#   2521    2522    2543    2564    2579    2593    2606    2621    2635    2711    2855    3241
                         4736    6513    7907
BIT6  = 000100           1421#   2494    2523    2544    2565    2580    2594    2607    2622    2636    2712    2857    3242
                         4741    4774    5758    5945
```

```
BIT7   = 000200          1420#  2495  2524  2545  2566  2567  2581  2595  2608  2623  2637  2714  3243
                         3592   3786  3799  4951  5091
BIT8   = 000400          1419#  2496  2525  2546  2547  2568  2582  2596  2609  2624  2638  2715  2858
                         3244   6517
BIT9   = 001000          1418#  2497  2526  2548  2583  2597  2610  2625  2639  2716  2859  3245  5080
                         6521   8804
BKSLSH  012107           3486#  5380  5388
BLDEXH  034432           6764   6794  7077#
BPTVEC= 000014           1434#
BSE    = 000200          2545#  6011  6809  7260  7426  7662
BSERR  = 000002          3237#  6193  6811  6825  6923
BSFACT  004162           2953#  5979  6027  6261
BSSOFT  003162           2952#  5991  6060  6266  6268
BUFFO   005212           2958#  3632  3636  3638  3643  3654  3656  3658  3704  3707  3709  3722  3724
                         3727   3729  3770  4344* 4345* 4347* 4349  4352  4357  4362* 4379* 4384  4388*
                         4817   4821  5047  5059  5119  5131  5383  5397* 5402* 5403  5407* 5485  5489
                         5499
BUFPRT  005422           2961#
B.BA16= 000001           2510#  7989
B.BA17= 000002           2511#  7989
B.CDT  = 000004          2512#  5501  7643  7989
B.CFMT= 000020           2513#  4984  5082  5456  5969  7033  7643  7989  8073
CART    011710           3460#  5466
CCLR   = 100000          2505#  7596
CDT    = 002000          2498#  3822  4924  7402  7952  8009  8060  8083
CERR   = 100000          2504#  6773  6803  7298  7357  7416  7598  7649  7964  8020  8108
CFMT   = 010000          2501#  7056  7089  7402  7952  8009  8060  8083
CKMEM1  026070           5882#
CKMEM2  026102           5886#
CKMO    026110           5885   5888#
CKSCOR  023176           5268#  6196
CLEAR  = 000105          2464#
CLR    = 000040          2521#
CLRPRM  021246           4869   4888#
CMDTO  = 000100          2712#  6658  6988  7159
CMPBUF  025022           5678#
CMSTRT  012206           3510   3515#  6914
CNTRDY  011507           3435#  5812
CNTRLC  012052           3480#  4740  5351
CNTRLG  012076           3484#  4754  5369
CNTRLR  012064           3482#  4750
CNTRLU  012071           3483#  5363
CNTRLZ  012057           3481#  4746  5357
COE    = 001000          2548#  6730  7260  7426  7662
COMAVG  007430           3210#  4388  4630* 4642* 4643* 4670
COMMA   012111           3487#  3682
COMSTR  005176           2956#  6325  6334  6942
CONCLR= 000176           2482#  8091
CONERR  030756           2843   6486#
CR     = 000015          1342#  8304  8314
CRLF   = 000200          1343#  8275  8314
CRTSER  024034           5090   5455#
CR2LF   012103           3485#  6392  6404  6428
CTLOUT  025542           4333   5785#  6306
CTO    = 004000          2499#  6693  7658
CYLNDR  005434           2966#
```

```
C.INIT  040210          6311    7869#
C.RTRN  042062          7973    8001    8029    8036    8039    8046    8050    8067    8089    8094    8104    8113    8126
                        8130    8135#
C.SPEC  041102          7909    8005#
DCEFLG  003144          2930#   4720*   6188*
DCK   = 100000          2554#   4716    6011    6827    7260    7426    7662
DCKERR= 000020          3240#   4719    5271    6128    6193    6829    6923
DCKHDL  020526          4711#   6190    6876    6896
DCLO  = 000020          2563#
DDISP = 177570          1349#   1526    3542
DDT   = 000400          2568#
DERCNT  003130          2918#   6757    6787    6836    6847
DESL  = 000010          2518#
DFLTDR  013744          3602    3784#
DFSTRT  012146          1453    3505#   5053    5125
DF01    052374          1657    1663    1669    1675    1717    1825    2095    2131    9485#
DF02    052424          1681    1699    1705    1711    1723    1729    1735    1741    1747    1753    1759    1765    1771
                        1777    1783    1789    1843    1849    1855    1963    1969    1975    1981    1987    1993    1999
                        2005    2011    2017    2059    2077    2101    9499#
DF03    052460          1687    1693    2065    2071    9514#
DF04    052520          1819    9531#
DF05    052534          1795    1801    9538#
DF07    052570          1807    9553#
DF10    052624          1813    9568#
DF11    052650          2113    9579#
DF12    052670          1861    1867    1873    1879    1885    1891    1897    1903    9588#
DF13    052720          1909    1921    9601#
DF14    052750          1915    9614#
DF15    053000          1927    9627#
DF16    053044          1933    9646#
DF17    053110          1951    9665#
DF20    053134          2119    9676#
DF21    053144          2029    9681#
DF22    053154          2125    9686#
DF23    053204          2035    2041    2047    9699#
DF24    053214          2053    9704#
DF25    053220          1957    5680*   5710*   5892*   5910*   9707#
DF26    053224          2107    9710#
DF27    053234          2083    9715#
DF30    053260          2137    9726#
DF31    053304          2143    9737#
DH100   050500          1655    1661    1667    1673    1679    1685    1691    1697    1703    1709    1715    1721    1727
                        1733    1739    1745    1751    1757    1763    1769    1775    1781    1787    1793    1799    1805
                        1811    1817    1823    1841    1847    1853    1859    1865    1871    1877    1883    1889    1895
                        1901    1907    1913    1919    1925    1931    1949    1961    1967    1973    1979    1985    1991
                        1997    2003    2009    2015    2057    2063    2069    2075    2081    2093    2099    2129    2135
                        2141    9302#   9688
DH101   050704          6409    9327#   9488    9501    9516    9533    9540    9555    9570    9590    9603    9616    9629
                        9648    9667    9690    9717    9728    9739
DH102   050773          6420    9337#   9490    9503    9518    9535    9542    9557    9572    9592    9605    9618    9631
                        9650    9669    9692    9719    9730    9741
DH103   051011          2117    9340#
DH104   051034          9344#   9678    9696
DH105   050522          6407    9305#
DH106   051054          9347#   9694
DH200   050545          9309#   9492    9505    9520    9594    9633    9652
```

DH201	051076	9350#	9494	9507	9522	9596	9635	9654					
DH202	051165	9360#	9509	9526	9637	9656							
DH203	051202	9363#	9511	9528	9643	9662	9712						
DH204	052065	9444#	9598										
DH500	050567	9313#	9496										
DH501	050625	9319#	9524	9641	9660								
DH502	052123	9450#	9611	9624	9658								
DH503	052164	9456#	9639										
DH601	051276	9373#	9544	9607	9620								
DH602	051320	9376#	9548										
DH604	051365	9383#	9559	9574	9581								
DH6041	051420	6030	9388#	9561	9576	9583							
DH6042	051447	2051	9392#										
DH605	051465	2027	9395#	9563									
DH6051	051502	9398#	9565	9683									
DH606	051537	9403#	9546	9550	9609	9622							
DH607	051565	9407#											
DH701	051605	5679*	5709*	5891*	5909*	9410#	9585						
DH702	051674	2105	9420#	9673	9723								
DH703	051710	9422#	9721										
DH704	051734	9426#	9734										
DH705	051754	9429#	9732										
DH706	051770	9431#	9743										
DH707	052003	9433#											
DH710	052033	9438#	9745										
DH711	052051	9441#	9671										
DH800	052241	9464#	9701										
DI =	040000	2503#	7446	8118									
DIF100	003147	2933#	4077*	4113*	4154*	4226*	4249*	4467	4472*				
DISPAK	010354	3319#	5106*	5107*	5110								
DISPLA	001142	1526#	3542*	3550*	8691*	8851*							
DISPRE	000174	1450#	3550										
DLT =	100000	2532#	6750	7254	7384	7420	7660						
DLTFLG	003146	2932#											
DMCART	022450	3998	4395	5104#									
DMD =	000040	2579#											
DONE	003132	2920#	6307*	6314	6353*	6487*	6605*	6957*	6960*				
DRA =	000001	2558#	7249	8032									
DRDY =	000200	2567#											
DRIDON	010612	3347#	5112										
DRIRDY	010573	3344#	5040										
DRIV	011701	3458#	5424										
DRIVE	005430	2964#	3814*	3817	4691	4872	4873	4918	4947	4961	4969	4992	5063*	5064*
		9469											
DRNAFG	003141	2927#	6308*	6500									
DRNAVL=	010000	3248#											
DROPDR	011611	3448#											
DROT =	000040	2564#	6740										
DROVFL	010215	3302#	3668										
DRPAR =	000010	2541#	6680										
DRPDRV=	002000	2719#	7894										
DRVBUF	011200	3393#	4340*	4342*	4343								
DRVCAL	030124	3839	3916	3919	4081	4172	4175	4470	4473	4503	4509	4512	4871	4941
		4976	4978	4986	5013	5071	5423	5465	5801	5842	5982	5999	6192	6304#
		6894	6941	6951	6987	7024	7026							
DRVDSC=	000040	2711#	6857	7334	7500	7562								

```
DRVERS  003137          2925#   3562*   6377*
DRVHRD= 000020          2710#   6852    7341    7536    7667
DRVLST  005512          2984#   3615    3620    3666    3730    3734    3801    4011    4257    4277    4330    4409    4627
                        5033    5156
DRVMSK= 000007          2517#   7243    7284    7435    7880
DRVNO   011675          3457#
DRVPDT= 000004          2708#   7442    7510
DRVPOS= 000002          2707#   7442    7469    7931
DRVPTR  005554          2988#   3739*   3804*   3999*   4000    4011*   4324    4336    4375    4396*   4397    4409*   5030
                        5063    5105
DRVSEQ  011235          3401#
DRVSER  023714          5089    5420#
DRVSZD= 010000          2721#   6671    7272    8034
DRVTST  013704          3695    3769    3774#
DRVUSE= 000001          2706#   7894
DRY   = 000200          2566#
DR.CLR= 000005          2489#   7322    7485    7527    7551    7665
DR.SEL= 000001          2488#   7392    7471    7786    7793    7800    7807    8011
DROXDP  011560          3443#
DSC   = 040000          2571#
DSWR  = 177570          1348#   1525    3541
DTBAII= 100000          2724#   4090    4235    6136    7894    7993
DTE   = 010000          2551#   6011    6745    7662
DTYE  = 000040          2543#   6708    7662
DT100   052264          1656    1662    1668    1674    1680    1686    1692    1698    1704    1710    1716    1722    1728
                        1734    1740    1746    1752    1758    1764    1770    1776    1782    1788    1794    1800    1806
                        1812    1818    1824    1842    1848    1854    1860    1866    1872    1878    1884    1890    1896
                        1902    1908    1914    1920    1926    1932    1950    1962    1968    1974    1980    1986    1992
                        1998    2004    2010    2016    2058    2064    2070    2076    2082    2094    2100    2124    2130
                        2136    2142    9469#
DT102   052302          9472#
DT201   052306          9473#
DT202   052324          2106    9476#
DT203   052330          9477#
DT601   052350          2028    2034    2040    2046    2052    2112    9480#
DT602   052356          1956    9481#
DZR6Q   007572          3251#   3566
ECCNC = 000040          3241#   6832
ECCW  = 020000          2587#
ECH   = 000100          2544#   6830    7260    7426    7662
ECOBAD  020764          4788#
EMTVEC= 000030          1437#   3526*   3527*
EM1     045405          1654    9002#
EM10    045620          1696    9029#
EM100   047731          2044    9236#
EM101   047754          2050    9240#
EM102   050032          2056    9248#
EM103   050104          2062    9255#
EM104   050132          2068    9259#
EM105   050154          2074    9262#
EM106   050207          2080    9267#
EM107   050242          2086    9272#
EM11    045627          1702    9031#
EM110   050307          2092    9279#
EM111   050342          2098    9284#
EM112   050363          2104    9287#
```

EM113	050416	2128	9292#	
EM114	050444	2134	9296#	
EM115	050461	2140	9299#	
EM12	045642	1708	9033#	
EM13	045663	1714	9036#	
EM14	045705	1720	9040#	
EM15	045733	1726	9044#	
EM16	045754	1732	9047#	
EM17	045771	1738	9050#	
EM2	045431	1660	9006#	
EM20	046012	1744	9053#	
EM21	046027	1750	9056#	
EM22	046047	1756	9059#	
EM23	046071	1762	9063#	
EM24	046122	1768	9068#	
EM25	046142	1774	9071#	
EM26	046165	1780	9075#	
EM27	046177	1786	9077#	
EM3	045455	1666	2146	9010#
EM30	046222	1792	1912	9081#
EM31	046247	1798	1918	9085#
EM32	046270	1804	9088#	
EM33	046311	1810	9091#	
EM34	046333	1816	2116	9095#
EM35	046353	1822	9098#	
EM36	046407	1828	9103#	
EM37	046442	1834	9108#	
EM4	045500	1672	9014#	
EM40	046505	1840	9114#	
EM41	046551	1846	9121#	
EM42	046607	1852	9127#	
EM43	046637	1858	9132#	
EM5	045521	1678	9017#	
EM52	046674	1900	9137#	
EM53	046722	1906	9141#	
EM56	046747	1924	1930	9145#
EM6	045540	1684	9020#	
EM60	046766	1936	9148#	
EM61	047022	1942	9153#	
EM62	047070	1948	9160#	
EM63	047114	1864	1960	9164#
EM64	047161	1870	1966	9171#
EM65	047224	1876	1972	9177#
EM66	047252	1882	1978	9181#
EM67	047305	1888	1984	9186#
EM7	045564	1690	9024#	
EM70	047344	1894	1990	9192#
EM71	047413	1996	9199#	
EM72	047453	2002	9205#	
EM73	047522	2008	9212#	
EM74	047556	2014	9217#	
EM75	047576	2020	9220#	
EM76	047626	2026	9224#	
EM77	047710	2032	2038	9233#
ENDTST	011140	3387#	4408	
EQUALS	012134	3496#		

```
ERRCNT  003135          2923#   3563*   5010*   6191*   6356*   6358    6361    6363*   6488*   6606*   6777*   6822*   6912*
                        6932    6935    6952*
ERRCOM  005426          2963#   6543*   6544*   9469
ERRFRE  030250          2841    4865    6353#   6880
ERRHDL  031472          2842    3840    4686    4718    4864    4944    5798    6006    6013    6205    6604#   6879    6899
ERRLMT  003136          2924#   6932
ERRVEC= 000004          1430#   3539    3540*   3551*   3559*   3560*   8829    8830*   8832*   8835*
ER2ENT  031512          6608#   6670    6783    6808    6904
E.CCLR= 000001          2850#   7601
E.CERR= 001000          2859#   7675
E.CLAT= 000020          2854#   7327    7491    7532    7556
E.CMTO= 040000          2861#
E.CONT  003062          2844#   6490    6494    6498    6505    6509    6513    6517    6521    6525    6529    6534*   7226*
                        7264*   7327*   7337*   7408*   7430*   7453*   7459*   7491*   7532*   7556*   7601*   7653*   7675*
                        7814*   8048*   8102*   8110*   8123*
E.DLT = 000400          2858#   7408
E.DPAR= 002000          2860#   7337    7814
E.ILLD= 000100          2857#   8102
E.MDS = 100000          2862#   7226    7653    8048
E.NOAT= 00C002          2851#   7453
E.SCLR= 000040          2855#   8123
E.UATT= 000004          2852#   7459
E.UDAT= 000010          2853#   7264    7430
E.UNLD= 020000          2722#   7345    7465    7540    7550
FACTBS  011732          3464#   6028
FINADR  024520          5585#   5715    5900
FINCYL  005470          2976#   5615*   5717    5735    5741    5902    5922    5928
FINMEM  027160          6072#   6123
FINSEC  005473          2978#   5603*   5719    5739    5743    5904    5926    5930
FINTRK  005472          2977#   5613*   5718    5737    5742    5903    5924    5929
FMTE  = 000020          2542#   6712    7662
FORMAT  003134          2922#   4092    4237    4874    5078*   5082*   5457    5595    5974    5994    6099    7034
GETPRM  020776          3743    3748    3757    4803#   4843
GETTYP  024154          3587    5477#   5480    5481    5482    5487
GNS   = ****** U        1449    8987    8988    8989    8990    8991    8994    8995    8996    8997
GO    = 000001          2493#   3820    3822    4921    4924
GTPKAD  034276          5274    6838    6849    7044#
GTPRMS  027516          6165#   6202
GTSWRG  021106          3584    4839#   5370    5818
GTO     027530          6164    6168#
HCRC  = 000400          2546#
HDR.AD  003100          2884#   7378    7382*   8072*
HDR.CT  003102          2885#   7386*   8075*   8078*
HIGOCT  005420          2960#
HLPOVL  003152          2937#   3565*
HLTPRG  033530          5813    5990    6910#   6938
HLTRQD  011537          3440#   5811
HOLD1   024364          5072    5493*   5503*   5511#   7027
HOLD2   024366          5075    5494*   5504*   5512#   7030
HOLD3   024370          4097    4112    4185    4248    5495*   5505*   5513#
HOLD4   024372          4233    5496*   5506*   5514#
HOLD5   024374          4469    4471    5497*   5507*   5515#
HT    = 000011          1340#   8273    8314
HVRC  = 000400          2547#   4712    6011    6784    7260    7426    7662
HVRCER= 000004          3238#   4714    5271    6193    6786    6923    7049
H.HEAD  003105          2888#
```

```
IDAE   = 002000          2549#   6735
IE     = 000100          2494#   3816    4917    4955    5037    5109    6769    6799    7606    7670    7958    7999    8065
                         8132
ILC    = 000001          2536#   6696    7662
ILF    = 000004          2539#   6704
INCRSC  027244           6084    6098#   6131
INITSS  021140           3815    3835    3869    3892    3940    3995    4040    4074    4146    4223    4320    4864#   4916
                         4937    5036    5061    5069    5083    5108    5133    5421    5455    5968
INTBLD  022620           3942    3962    4109    4232    5146#
INTCHR  005442           2969#   4737*   4756    4773*   4788    5094*   5787    5791    5796*   5805    5809    5816
INTMSK  003111           2893#   7301    7325    7346    7371    7457    7489    7530    7541    7554    7825*   7829*   7833*
                         7882*   7968
INTR   = 000300          2484#
IOTVEC= 000020           1435#   3524*   3525*
IR     = 000100          2523#
I.AERR  036572           7477    7483    7501    7510#
I.ATTN  036122           7237    7281    7414#
I.CCLR  037156           7315    7365    7596#   7668    8028    8093
I.CSTS  037610           7308    7743#
I.CST1  037632           3830    4932    7312    7364    7747#   8017
I.DRV   003112           2897#   7882    7883
I.ERR   035344           7309    7313#
I.ERRA  035324           7308#   7376
I.ERRC  035332           7246    7299    7311#   7444
I.HDAL  035570           7292    7355#
I.INTR  034710           3574    7216#
I.ISRL  003104           2887#   7230    7233*   7236*   7607*   7904*   8058*   8131*
I.ISSU  037240           7323    7393    7472    7486    7528    7552    7638#   7787    7794    7801    7808
I.IOO   035170           7234    7251    7278#
I.RTRN  037140           7228    7266    7276    7304    7316    7321    7324    7330    7339    7348    7351    7366    7368
                         7394    7397    7406    7410    7432    7448    7455    7461    7473    7487    7494    7506    7526
                         7529    7534    7543    7546    7553    7558    7567#
I.STAT  037714           7320    7525    7784#   8038
I.STOR  037462           7648    7698#
I.UNLD  037040           7467    7550#
KBDHDL  020604           3571    4729#
LASTWC  005474           2979#   5586    5714*   5899*
LDMEM1  026020           5854#
LDMEM2  026032           5858#
LDMO    026040           5857    5860#
LEV2ER= 000400           3244#   6608    6663    6760    6766    6790    6796    6900    6958
LF     = 000012          1341#   8308    8314
LODBUF  024676           5633#
LODSEC  024460           3896    4159    5557#
LOGDRV  003154           2939#   3898    4084    4107    4156    5035*   5230    5236
LOWOCT  005416           2959#
LSTCYL  024362           4980    5461    5492*   5502*   5510#   5971
LSTTRK= 000002           3236#   4981    5462
MA      005500           2981#
MCLK   = 000400          2582#
MDFLAG  003126           2916#   3505*   3508*   3512*   3588    3693    4012    4410    5051    5055    5085    5123    5127
                         5789
MDS    = 001000          2526#   3826    4928    7224    7651    8026
MEMABT  003151           2935#
MEMVEC= 000114           2950#
MERD   = 001000          2583#
```

```
MESMSK= 000017           2576#
MEWD  = 002000           2584#
M!DXFR  027310           6122#   6197
MIND  = 000200           2581#
MNTPAK  010310           3313#   5031*   5032*   5038
MSP   = 000100           2580#
MTCART  022066           3871    4043    5029#
M.CADD= 017760           2652#
M.CDIF= 017760           2651#
M.DPID  042664           4598    4663    8357#
M.DPIM  042624           8328#
M.DP01  042634           8331#   8340
M.DP02  042654           8335    8339#
M.DP40  042722           8368#   8389
M.DP41  042756           8373    8380#
M.DP42  042766           8379    8384#
M.DP44  043020           8391    8396#
M.DP50  043032           8366    8400#
M.DRV = 000007           2648#
M.HEAD= 007000           2655#
M.ID  = 000003           2650#
M.PAR = 100000           2649#
M.SECT= 000760           2654#
M.SER = 077770           2653#
NCSCR0  007112           3162#   4227    4229    4255    4272    4550    4585    4632
NCSCR1  007152           3165#
NCSCR2  007212           3168#
NED   = 010000           2529#   4684    6642    6652    7239    7317    7660    8030
NEDHDL  020450           3837    4684#   4939
NEM   = 004000           2528#   6637    7254    7420    7660
NEWMSG  011224           3399#   4807
NEWON   005476           2980#   3838*   3841    4696*   4940*   4945
NGOFST  005452           2973#
NOCHK = 000400           2715#   7894    7956    7997    8024    8042    8063    8086    8095    8129
NODRLS  010622           3349#   3736
NODRTS  011456           3430#
NODSC = 004000           2720#   6862    7482
NOFALS  011777           3471#   6037
NORTRY  003150           2934#   4868*   6928
NOSCR0  006612           3140#   4102    4104    4328    4546    4580    4631    4648
NOSCR1  006652           3143#
NOSCR2  006712           3146#
NOTMSG  054330           3569    9750#
NOTYPE  003153           2938#   3567    3570*
NSCOR0  007432           3215#   4163*   4461    4463    4485    4491    4533    4553    5298
NSCOR1  007452           3218#   4164*   4535
NSCOR2  007472           3221#   4165*   4537
NSSCR0  007412           3185#   4148*   4256    4534*   4575
NSSCR1  007414           3188#   4149*   4536*
NSSCR2  007416           3191#   4150*   4538*
NTREDY  011303           3409#   4964
NXDRIV  011264           3406#   4950
NXF   = 000004           2540#
OAVERG  011102           3381#   4380
OCTBIN  042104           4822    8160#
OFFSET= 000115           2468#   4502    4511    7939
```

```
OFINUS  005436           2967#
OFSCNT  003157           2942#  4477*  4496   4504*  5309
OFSDIR  003156           2941#  4474*  4486   4493   4518*  4589*  4595   4604*  5299
OFST  = 000004           2560#
ONSUBS  010124           3292#  3688*  3689
OPCOMP  003131           2919#
OPCONT  003140           2926#
OPI   = 020000           2552#  6011   6754   7260   7426   7662
OPIERR= 000010           3239#  6756   6923
OR    = 000200           2524#
OREQSW= 000020           2947#
OVRAVG  007426           3207#  4379   4629*  4640*  4641*  4657
O.WAIT  003064           2864#  7160   7164*  7274*  7278   7294*  7360*  7373*  7388*  7932*  7992*  8054*  8071*
PACK  = 000103           2463#  4975
PARM0   002640           2753#  2908   4867   5011   6158   6167   6875   6891   6895*  6949   7001   7035
PARM1   002724           2785#  4873*  4876*  6610   6891*  6895   6985   7019
PASSNO  003155           2940#  3509*  3513*  3774   3776   4042*  5087   5814*
PAT   = 000020           2578#
PATRN   005446           2971#  5635   5684   5760
PATOO   005566           2991#  5637   5652   5686   5701
PBLKT   003122           2908#  7157   7244   7270   7438   7463   7881*
PBSVAL= 001000           2716#  7819
PCA   = 004000           2585#
PCD   = 010000           2586#
PCSCR0  007252           3173#  4275   4292   4551   4586
PCSCR1  007312           3176#
PCSCR2  007352           3179#
PGE   = 002000           2527#  6700   7660
PIP   = 020000           2570#
PIRQ  = 177772           1347#
PIRQVE= 000240           1441#
PLOFST  005450           2972#
PMA     005504           2982#  5654   5703   5752   5754   5861   5889   5939   5941   6141*  6142*  6143*  6144
                         6145   6162*  6163*
POSCR0  006752           3151#  4547   4581
POSCR1  007012           3154#
POSCR2  007052           3157#
PREPKB  020744           3870   3893   3941   3996   4041   4075   4147   4224   4321   4772#  5819
PRMBUF  012122           3494#
PRMPSP  012143           3498#
PROMPT  012140           3497#  4354
PRVCMD  005162           2954#  3578   6324   6577
PR0   = 000000           1364#  3577   5014   6950   7023
PR1   = 000040           1365#
PR2   = 000100           1366#
PR3   = 000140           1367#
PR4   = 000200           1368#  3572
PR5   = 000240           1369#  2840
PR6   = 000300           1370#
PR7   = 000340           1371#  3515   8784   8785
PS    = 177776           1344#  1345   3515*  3577*  5014*  7146   7152*  7166*  7875   7876*  8135*
PSCOR0  007512           3226#  4166*  4495   4539   4554   5301
PSCOR1  007532           3229#  4167*  4541
PSCOR2  007552           3232#  4168*  4543
PSSCR0  007420           3196#  4151*  4276   4540*  4576
PSSCR1  007422           3199#  4152*  4542*
```

```
PSSCR2  007424          3202#   4153*   4544*
PSW    = 177776         1345#   6950*   7023*
PWRMSG  044526          8787    8791#
PWRVEC= 000024          1436#   3530*   3531*   8775*   8783*   8784*
P.ASOF= 000032          2737#   6564    7488*   7489    7522*   7753*   7941*   7942
P.A00 = 000040          2740#   4959    4967    6567    6991    7479*   7480    7496*   7497    7810*   8018*
P.A01 = 000044          2742#   6569    7789*
P.A10 = 000050          2744#   6571    7796*
P.A11 = 000054          2746#   5426    6573    7803*
P.BAHI= 000007          2699#   6147*   6553
P.BALO= 000010          2700#   3912*   4089*   4158*   4234*   4982*   5463*   5838*   5979*   5991*   6144*   6556    7975
                        8072
P.BAR = 000024          2734#   6563    7519*   7749*
P.B00 = 000042          2741#   6568    7474*   7475    7811*   8019*
P.B01 = 000046          2743#   6570    7790*
P.B10 = 000052          2745#   6572    7013    7797*
P.B11 = 000056          2747#   6574    7395*   7804*
P.CMND= 000001          2693#   3828*   3836*   3915*   3918*   4079*   4083*   4171*   4174*   4466*   4502*   4505*   4508*
                        4511*   4670*   4930*   4938*   4975*   4977*   4979*   5012*   5070*   5422*   5460*   5800*   5841*
                        5970*   6544    6892*   6940*   6986*   7022*   7025*   7290    7907    7911    7923    7933    7939
                        7945    7948    7954    7979    7986    8005    8052    8069    8091    8100
P.CS1 = 000016          2731#   3829*   4891    4931*   6557    6675    6693    7056    7089    7311*   7363*   7401*   7402*
                        7404*   7405    7514*   7516*   7743*   7746*   7897    7951*   7952*   7954*   7958*   7959    7983*
                        7986*   7988*   7989*   7999*   8000    8008*   8009*   8011*   8012    8014*   8015    8020    8059*
                        8060*   8062*   8065*   8066    8082*   8083*   8085*   8088    8107*   8108    8117*   8118
P.CS1H= 000007          2698#   4874*   4875*   4876*   4984*   4985*   5456*   5969*   7033*   7034*   7401    7642    7951
                        7988    8008    8059    8073    8082
P.CS2 = 000020          2732#   4684    6558    6632    6637    6642    6644    6650    6652    6700    6750    6843    7317
                        7517*   7747*   7905*   7930    7995*   7996    8007    8026    8030    8056*   8057    8081
P.CYLN= 000002          2694#   3903*   3908*   3946*   3949*   3952*   3955*   3965*   3968*   4080*   4085*   4097*   4108*
                        4112*   4170*   4173*   4241*   4244*   4469*   4471*   4980*   5247*   5461*   5614    5971*   6073
                        6132*   6546    7021*   7399    7935    7977    8079
P.DCYL= 000030          2736#   6077    6124    6559    6812    7044    7078    7521*   7755*
P.DRVN= 000000          2692#   4872*   7287    7464    7640    7879
P.DS  = 000036          2739#   6565    6685    6689    6740    7524*   7751*   8032
P.DTS = 000026          2735#   6079    6081    6125    6126    6560    6813    6815    7012    7047    7048    7079    7085
                        7520*   7750*
P.EPAT= 000062          2749#   6840    7313*   7899
P.EPOS= 000060          2748#   6839    7314*
P.ER  = 000034          2738#   4712    4716    6011    6566    6631    7523*   7752*
P.OFST= 000006          2697#   4501*   4510*   7941
P.PRST= 000014          2702#   4090*   4235*   6136    6658    6671    6852    6857    6862    6867    6988    7159*   7272*
                        7306*   7334*   7341    7345*   7442*   7465    7469*   7482*   7500*   7510*   7536    7540*   7550*
                        7562*   7667*   7819*   7894*   7931*   7955*   7956    7981    7987*   7993    7997    8024    8034*
                        8042    8063    8086    8095    8129
P.SECT= 000004          2695#   3905*   3910*   4161*   4182*   4183    5240*   5459*   5591    5976*   5985*   5986    5992*
                        6002*   6003    6075    6134*   6549    7020*   7400    7936    7978    8080
P.TRCK= 000005          2696#   3921*   3922    4162*   4178*   4179    4480    4981*   5241*   5243*   5244    5462*   5605
                        5972*   6074    6133*   6547
P.WC  = 000012          2701#   3913*   4091*   4094*   4160*   4236*   4239*   4983*   5464*   5839*   5978*   6135*   6160
                        6198    6551    7976
P.WCR = 000022          2733#   6561    7518*   7748*
P1SET   012162          3506    3509#
P1STRT  012154          1455    3508#
P2STRT  012172          1457    3512#
QNEWSW= 000000          2452
```

```
Q.INIT= 040000          2723#
RAVERG  011121          3384#   4389
RCDASW= 000010          2946#
RCLREQ= 004000          3247#   6678    6683    6728    6733    6738    6743    6780    6889
RDALHD= 000164          2481#   7290    8069
RDCHR = 104406          5346    8994#
RDCHRS  023402          3624    3698    3716    4808    5041    5113    5340#   5479
RDDATA= 000121          2470#   4505    4979    5460    5970
RDGATE= 100000          2589#
RDHDO   034122          6765    6795    7010#
RDHEAD= 000125          2472#   5070    7025    7404    8085
RDSTAT= 000141          2480#   3836    4938    5422    6986    8005
RDY   = 000200          2495#   5073    5076    6885    7028    7031    7962    8015
RECAL = 000113          2467#   4977    5800    6892    7948
RECODE  005424          2962#   3833*   4711*   4714*   4719*   4935*   4987    5009*   5271    5722    5907    6128    6189*
                        6193    6270*   6278*   6354    6364*   6607*   6608    6635*   6640*   6661    6663    6678*   6683*
                        6723*   6728*   6733*   6738*   6743*   6748*   6756*   6760    6766    6780*   6786*   6790    6796
                        6811*   6818    6825*   6829*   6832*   6845*   6846*   6855*   6860*   6865*   6870*   6881    6887*
                        6889    6900    6902*   6923    6925*   6930    6937*   6958*   6990*   7049
REDBSF  026516          5084    5967#
REDOFS  017250          4110    4177    4242    4457#
REISSU  003142          2928#   6502*
RELEAS= 000140          2479#   8052
REPLPK  011367          3420#
REPSUP  031212          3831    4933    5683    5893    5988    6541#   6613
RESLTS  010667          3356#   4325*   4326*   4327
RESREG= 104410          4520    4561    4613    4673    5189    5215    5250    5312    5445    5567    5617    5664    5721
                        5770    5843    5869    5906    5956    6063    6088    6148    6172    6271    6341    6473    6592
                        6953    7002    7036    7097    8432    8484    8996#
RESVEC= 000010          1431#
RETANL  033724          6611    6893    6957#
RETNML  033742          4699    4721    6015    6612    6960#
RKASOF= 000016          2448#   6979    7300    7370    7450    7705    7753    7942*   7967
RKBA  = 000004          2443#   6978    7701    7749    7975*
RKBADR  007630          3256#   3742
RKBAS   003046          2838#   3591*   3595*   3741    3744*   4866    6768    6798    6884    6939    6971    7153    7222
                        7886
RKCS1 = 000000          2441#   4955*   5037*   5072*   5073    5075*   5076    5109*   6769*   6773    6799*   6803    6973
                        7027*   7028    7030*   7031    7245    7297    7355    7405*   7414    7596*   7597    7606*   7645*
                        7646    7670*   7698    7959*   7961    8000*   8012*   8014    8044*   8066*   8088*   8097*   8107
                        8117    8132*
RKCS2 = 000010          2445#   3817*   3826    4918*   4928    6974    7223    7383    7641*   7699    7747    7930*   7996*
                        8007*   8057*   8081*   8106*
RKDA  = 000006          2444#   6976    7400*   7702    7750    7936*   7978*   8080*
RKDB  = 000024          2451#   5079    5080    6770    6771    6772    6800    6801    6802    7379    7380    7381
RKDC  = 000020          2449#   6975
RKDCYL= 000020          2450#   7399*   7706    7755    7935*   7977*   8079*
RKDS  = 000012          2446#   6885    6980    7248    7703    7751
RKECPS= 000030          2456#   7314    7710
RKECPT= 000032          2458#   7313    7711
RKER  = 000014          2447#   6981    7257    7423    7704    7752
RKMR1 = 000026          2452#   7391*   7470*   7707    7784*   7791*   7798*   7805*   7906*
RKMR2 = 000034          2453#   7708    8018
RKMR3 = 000036          2454#   7709    8019
RKPAT = 000032          2457#
RKPOS = 000030          2455#
```

```
RKPRI   003052          2840#  3575   3600*  3751   3767*  7152   7876
RKPRTY  007677          3264#  3756
RKVADR  007654          3260#  3747
RKVEC   003050          2839#  3573   3596*  3597*  3746   3749*
RKWC  = 000002          2442#  6977   7700   7748   7976*
RLS   = 000010          2519#  8056
ROTBLD  023000          3960   4115   4247   5200#
RWBUF   053330          3912   4088*  4089   4158   4234   4982   4990   5463   5468   5558   5562   5639   5688
                        5832   5838   6162   9748#
R.ABNL  040142          7165   7275   7350   7367   7545   7565   7825#  8035
R.CONT  040166          7227   7265   7329   7338   7409   7431   7454   7460   7493   7533   7557   7602   7654
                        7677   7815   7833#  8049   8103   8112   8125
R.NORM  040154          7303   7396   7505   7829#  7972   8045   8133
SAVPRS  005454          2974#  6159   6166
SAVREG= 104407          4457   4530   4572   4625   5146   5200   5229   5268   5420   5557   5585   5633   5678
                        5716   5830   5854   5858   5882   5886   5901   6025   6072   6122   6157   6165   6260
                        6323   6376   6486   6542   6604   6970   7011   7077   8417   8458   8995#
SAVWRD  003160          2951#
SCLR  = 000040          2522#  8106
SCNDRV  021274          4916#  5065
SCOPER= 104411          8997#
SCOPE1  044550          8802#  8997
SCRACH  005444          2970#  3675*  3678*  3683   3823*  3824*  4925*  4926*  5381*  5383*  5384   5394*  5395*
                        5396   5682*  5724   5730*  5733   5890*  5911   5917*  5920
SCRLST  005556          2989#  3648   3656*  3662   3678   3790*  3793*  3794*  3812   3847*
SECLST  006572          3130#  5240   5287
SEEK  = 000117          2469#  4079   4171   4466   7923   7933
SELDRV= 000101          2462#  3828   4930   8062
SELF    011205          3394#  4338
SERNM   011717          3462#  5425   5467
SETTUP  022020          3868   3891   3939   3994   4039   4073   4145   4222   4319   5007#
SKI   = 000002          2538#  6725
SOFTBS  011745          3466#  6061
SPACE1  012120          3492#  4351
SPACE2  012117          3491#  4356   6042   6048   6415   6419   6422   6425   6445   6450   6465
SPACE3  012116          3490#
SPACE4  012115          3489#
SPACE6  012113          3488#
SPAR  = 020000          2502#  6675   7658
SPDLSS= 000020          2562#  6689
SRHTBS  027656          6221#  6262
SRPASS  010257          3308#  3774*  3775   4014*  4015
SRTSPL= 000111          2466#  7945
STACK = 001100          1335#  3522   5007*  5008   5050   5057   5122   5129   5799   8786
STALL   024376          5531#  6305
STALLS  005432          2965#  3583*  5540
STDLST  006552          3120#  5148
STKLMT= 177774          1346#
STRCMD  030176          6309   6323#
STR204  01042C          3325#  4007*  4008
STR220  010475          3333#  4017*  4018   4404*  4405
SUBCLR= 000177          2483#  4508   4870   5012   6940   7022   8100
SUBSYS  005510          2983#  3601*  3729*  3731   3802   4003   4400
SVAL  = 100000          2572#
SVPRMS  027456          4095   4506   6157#  6203
SWR     001140          1525#  3520   3541*  3543   3549*  3556*  4839   5091   5533   5731   5758   5918   5945
```

```
                        6378    6405    6433    8692    8699    8711    8714    8804    8824    8840
SWREG   000176            1451#   3549    4839    4841    4844*
SWRMSG  011215            3397#   4842
SW0    = 000001           1399#
SW00   = 000001           1389#   1399
SW01   = 000002           1388#   1398
SW02   = 000004           1387#   1397
SW03   = 000010           1386#   1396
SW04   = 000020           1385#   1395
SW05   = 000040           1384#   1394
SW06   = 000100           1383#   1393
SW07   = 000200           1382#   1392
SW08   = 000400           1381#   1391
SW09   = 001000           1380#   1390
SW1    = 000002           1398#
SW10   = 002000           1379#
SW11   = 004000           1378#
SW12   = 010000           1377#
SW13   = 020000           1376#   6378
SW14   = 040000           1375#
SW15   = 100000           1374#
SW2    = 000004           1397#
SW3    = 000010           1396#
SW4    = 000020           1395#
SW5    = 000040           1394#
SW6    = 000100           1393#
SW7    = 000200           1392#
SW8    = 000400           1391#
SW9    = 001000           1390#
SYSDRV  010053            3284#   3622*   3623
S.ACLO= 000100            2607#
S.BRHM= 000100            2622#
S.BRKE= 040000            2644#
S.CART= 000400            2624#
S.DCLO= 010000            2614#
S.DIB  = 002000           2640#
S.DOOR= 000200            2623#
S.DRA  = 000040           2593#
S.DROT= 020000            2615#
S.DRY  = 000200           2595#   4959
S.DSC  = 040000           2602#   7332    7480    7497    7560
S.FLT  = 000200           2608#   7475
S.FORM= 001000            2597#
S.FWD  = 002000           2626#
S.HDFL= 000200            2637#
S.HDHM= 000040            2621#
S.ICYL= 000040            2606#
S.ILF  = 000400           2609#
S.LIMD= 020000            2643#
S.LOAD= 010000            2628#
S.MHD  = 000400           2638#
S.NMOV= 010000            2642#
S.OFF  = 002000           2598#
S.PAR  = 001000           2610#   7335    7812
S.PIP  = 020000           2601#   7343    7538
S.PLO  = 004000           2641#
```

```
S.REV = 004000           2627#
S.RTZ = 020000           2629#
S.SECT= 000020           2634#
S.SKI = 002000           2611#
S.SPIN= 010000           2600#
S.SPLS= 010000           2613#
S.SPOK= 001000           2625#
S.TYPE= 000400           2596#
S.UNLD= 040000           2630#
S.UNS = 040000           2616#
S.VV  = 000100           2594#
S.WCLK= 000040           2635#
S.WGAT= 000100           2636#
S.WLE = 004000           2612#
S.WRL = 004000           2599#   4967
S.XDOK= 000020           2620#
S.XERR= 001000           2639#
TAB     011213           3396#   4360
TABLEA  005566           2997#   3903    3905
TABLEB  005766           3037#   5559
TABLEC  006026           3058#   3946    3949    3965
TABLED  006112           3079#   514'    5164    5178    5181    5201    5213    5233    5285
TABLEG  006512           3099#   4087    5232
TABL7A  005666           3017#   3908    3910
TABL7C  006060           3068#   3952    3955    3968
TBITVE= 000014           1432#
TKVEC = 000060           1439#   3571*   3572*
TOPROC  033752           6489    6660    6969#
TPVEC = 000064           1440#
TRACK   005440           2968#
TRAPVE= 000034           1438#   3528*   3529*   8785*
TRKAVG  020232           4374    4625#
TRKBUF  011174           3392#   4331*   4334
TRNSFR  027546           4096    4507    6186#
TRTVEC= 000014           1433#
TSTDRN  011654           3454#
TSTING  003127           2917#   3576*   3894*   3997*   4076*   4322*   4394*   5785    5797*
TSTMSG  045366           6399    8999#
TSTTYP  003143           2929#   3561*   4101*   4120*   4169*   4176*   4189*   4225*   4253*   4459    4478    4531    4548
                         4573    4583    5269    5275    5802*   6775    6820    6877    6897
TST1    014316           3778    3866#   4006
TST10   015766           4220#
TST11   016404           4317#
TST2    014346           3889#
TST3    014540           3937#
TST4    014732           3992#
TST5    015116           3780    4016    4037#   4403
TST6    015154           4071#
TST7    015460           4143#
TWOTOS= 002000           3246#   6661    6990
TYPBSF  027006           5093    6025#
TYPDS = 104405           8991#
TYPE  = 104401           3566    3569    3612    3623    3632    3633    3668    3674    3682    3683    3689    3697    3709
                         3710    3715    3724    3725    3736    3742    3747    3756    3768    3770    3771    3775    4008
                         4015    4018    4327    4334    4338    4343    4351    4354    4356    4360    4365    4368    4381
                         4382    4387    4392    4405    4408    4740    4746    4750    4754    4756    4757    4788    4789
```

```
                        4807    4817    4818    4842    4949    4950    4963    4964    4971    4972    4989    4994    4995
                        5038    5039    5040    5047    5048    5110    5111    5112    5119    5120    5351    5357    5363
                        5369    5380    5384    5388    5396    5401    5403    5404    5406    5424    5425    5444    5466
                        5467    5471    5477    5478    5485    5486    5811    5812    6028    6029    6030    6031    6037
                        6038    6042    6048    6052    6061    6392    6393    6394    6398    6399    6404    6407    6408
                        6409    6410    6415    6418    6419    6420    6421    6422    6425    6428    6429    6431    6445
                        6450    6454    6460    6465    6466    6468    8278    8525    8590    8665    8694    8702    8787
                        8987#
TYPERR  030330          6376#   8701
TYPFMT  003133          2921#   3818    3901    3944    3963    4875    4919    4985    5491*   5501*
TYPOC = 1 4402          6041    6047    6051    6414    6424    6427    6447    8988#
TYPON = 104404          8990#
TYPOS = 104403          4804    6401    8989#
TYPRWN  010552          3341#   5039    5111
TYPSYS  007722          3268#   3715
T.ASOF  003024          2825#   7300*   7301    7325    7370*   7371    7450*   7451    7457    7488    75?2    7530    7554
                        7705*   7967*   7968
T.BA    003016          2822#   7519    7701*
T.CS1   003010          2817#   7245*   7297*   7298    7311    7322*   7355*   7357    7363    7392*   7414*   7416    7446
                        7471*   7485*   7515*   7516    7527*   7551*   7597*   7598    7638    7642*   7643*   7645    7649
                        7658    7698*   7745*   7746    7786*   7793*   7800*   7807*   7961*   7962    7964
T.CS2   003012          2819#   7223*   7224    7239    7242    7254    7283    7383*   7384    7420    7434    7517    7639*
                        7640*   7641    7651    7660    7699*
T.DA    003020          2823#   7520    7702*
T.DB    003044          2834#
T.DC    003022          2824#   7521    7706*
T.DS    003030          2828#   7248*   7249    7524    7703*
T.ER    003026          2827#   7257*   7260    7423*   7426    7523    7662    7704*
T.MR1   003032          2829#   7707*
T.MR2   003034          2830#   7332    7343    7479    7496    7538    7560    7708*   7789    7796    7803    7810
T.MR3   003036          2831#   7335    7395    7474    7709*   7790    7797    7804    7811    7812
T.PAT   003042          2833#   7711*
T.POS   003040          2832#   7710*
T.WCR   003014          2821#   7518    7700*
UEXATT= 000010          2709#   6867    7306
UFE   = 000400          2525#   6644    6650    7239    7317    7660    8030
UNLOAD= 000107          2465#   7911
UNS   = 040000          2553#   6720
UPE   = 020000          2530#   6632    7254    7420    7660
VFHDSW= 000002          2944#
VV    = 000100          2565#
WCDASW= 000004          2945#
WCE   = 040000          2531#   6843    7254    7420    7660
WCEFLG  003145          2931#   5981*   5983    5998*   6000    6014*
WCERR = 000100          3242#   6846    6923
WDSXFR  005466          2975#   5634    5681    5713    5860    5888    5898    6087*   6127    6160*   6161*
WHDSW = 000001          2943#
WILTST  010100          3288#   3674
WLE   = 004000          2550#   6716    7260    7426
WORDSP  012126          3495#
WRDATA= 000123          2471#   3915    4083    4174    5841    7983
WRHEAD= 000127          2473#
WRL   = 004000          2569#
WRTBLK  023064          3947    3953    3956    3969    5229#
WRTCHK= 000131          2474#   3918    7979
WRTGAT= 040000          2588#
```

```
WRTLOK  011317            3412#   4972
WRTSEC  025746            5242    5830#
W.DRV   003124            2914#   7154*   7269*   7295*   7347*   7362*   7375*   7389*   7437*   7504*   7513*   7542*   7564*
                          7835*   7884*   7947*   7950*   7971*   8023*   8041*
W.MILI  003070            2867#   7149    8115
W.MIN   003076            2883#   7947
W.MTIM  003066            2866#   7147*   7149*
W.SEC   003072            2880#   7884
W.TIME  003110            2889#   7150    7156*   7268*   7296*   7346*   7361*   7374*   7390*   7436*   7503*   7512*   7541*
                          7563*   7834*   7883*   7970*   8022*   8040*
W.WCK = 000200            2714#   7894    7955    7981    7987
W.WTCH  034560            6313    7142#   7960    7966    8013
W.BSEC  003074            2882#   7347    7542    7950
$APTHD  001000            1482    1488#
$ASTAT= ****** U          8888    8903
$ATYC   044770            8859    8861#
$ATY1   044744            8857#
$ATY3   044752            8263    8858#
$ATY4   044762            8707    8860#
$AUTOB  001134            1522#
$BASE   001404            1633#   3591
$BDADR  001122            1517#
$BDDAT  001126            1519#
$BELL   001320            1584#   8694    8726
$CHARC  042620            8280*   8290*   8297    8306*   8311#
$CKSWR= ****** U          8994
$CMTAG  001100            1505#   3517    3518    3526    3532    3533
$CM1  = 000040            1537#   1538#   1539#   1540#   1541#   1542#   1543#   1544#   1545#   1546#   1547#   1548#   1549#
                          1550#   1551#   1552#   1553#   1554#   1555#   1556#   1557#   1558#   1559#   1560#   1561#   1562#
                          1563#   1564#   1565#   1566#   1567#   1568#   1569#
$CM2  = 000100            1537#   1538#   1539#   1540#   1541#   1542#   1543#   1544#   1545#   1546#   1547#   1548#   1549#
                          1550#   1551#   1552#   1553#   1554#   1555#   1556#   1557#   1558#   1559#   1560#   1561#   1562#
                          1563#   1564#   1565#   1566#   1567#   1568#   1569#
$CM3  = 000040            1535#   1537
$CM4  = 000016            1569#   1570#   1571#   1572#   1573#   1574#   1575#   1576#   1577#   1578#   1579#   1580#   1581#
                          1582#   1583#
$CNTLG  044411            8764#
$CNTLU  044404            8763#
$CPUOP  001356            1607#
$CRLF   001325            1586#   3612    3768    4365    4368    4387    4392    4757    5401    5406    5444    5471    5477
                          6031    6038    6052    6398    6408    6410    6418    6421    6431    6454    6460    6468    8279
                          8314    8702    8726
$DBLK   044060            8631    8665    8673#
$DB2D   043162            4358    4385    8458#
$DB2O   043042            5469    8417#
$DECVL  043342            8460    8506#
$DEVCT  001340            1598#
$DEVM   001406            1634#   3785
$DOAGN  017244            4432    4437    4443#
$DTBL   044050            8634    8669#
$ENDAD  017234            1468    4439#   8721
$ENDCT  017220            3532    4434#
$ENV    001350            1603#   8258    8704    8866    8890
$ENVM   001351            1604#   3554    3592    3786    3799    4951    8260    8265    8868
$EOP    017170            4425#
$EOPCT  017212            3532*   4431#   4435
```

```
$ERFLG  001103          1508#   8689*   8726    8802    8816    8838    8844*   8854
$ERMAX  001115          1514#   3534*   8850*   8854
$ERROR  044070          3526    8688#
$ERRPC  001116          1515#   8696*   8697*   8698    8726    9469
$ERRTB  001410          1651#   6389
$ERTTL  001112          1512#   8695*   8726
$ESCAP  001316          1583#   3533*   8717    8719    8726    8849*
$ETABL  001350          1602#
$ETEND  001410          1494    1635#
$FATAL  001332          1595#   8894*
$FFLG   042210          8857*   8860*   8888    8897*   8905#
$FILLC  001156          1533#   8283    8314
$FILLS  001155          1532#   8314
$GDADR  001120          1516#
$GDDAT  001124          1518#
$GET42  017224          4436#
$GTSWR= ****** U        8993
$HD   = 000002          1316
$HIBTS  001000          1489#
$HINUM  042336          3585*   8215    8223    8228*   8233#
$HIOCT  042236          4825    8185*   8198#
$ICNT   001104          1509#
$INTAG  001135          1523#
$ITEMB  001114          1513#   6384    8698*   8706    8726
$LF     001326          1587#   8314    8726
$LFLG   045207          8898*   8904#
$LONUM  042340          3586*   5536    8214    8221    8227*   8234#
$LPADR  001106          1510#   3535*   8789    8842*   8847*   8852    8854
$LPERR  001110          1511#   3536*   8716    8806    8842    8848*   8854
$MADR1  001362          1620#
$MADR2  001366          1624#
$MADR3  001372          1627#
$MADR4  001376          1630#
$MAIL   001330          1490    1494    1593#   3553    3867    3890    3938    3993    4038    4072    4144    4221    4318
                        8258    8704    8846
$MAMS1  001360          1614#
$MAMS2  001364          1622#
$MAMS3  001370          1625#
$MAMS4  001374          1628#
$MBADR  001002          1490#
$MFLG   045206          8858*   8864    8899*   8903#
$MNEW   044427          8767#
$MSGAD  001344          1600#   8874*   8877
$MSGLG  001346          1601#   8879*
$MSGTY  001330          1594#   8872    8880*   8892    8896*
$MSWR   044416          8765#
$MTYP1  001361          1615#
$MTYP2  001365          1623#
$MTYP3  001371          1626#
$MTYP4  001375          1629#
$NULL   001154          1531#   8285    8314
$NWTST= 000001          3854#   3856    3877#   3879    3928#   3930    3973#   3975    4024#   4026    4047#   4049    4124#
                        4126    4193#   4195    4297#   4299
$OCNT   043640          8562*   8591*   8604#
$OCTVL  043144          5427    8419    8444#
$OMODE  043642          8557*   8561*   8566    8569*   8580*   8606#
```

```
$OVER   044730           8825    8843    8851#
$PASS   001336           1597#   3553*   4428*   4429*   4445
$PASTM  001006           1492#
$PWRCT  044524           8780*   8781*   8790#
$PWRDN  044440           3530    8775#   8783
$PWRUP  044452           8775    8780#
$QUES   001324           1585#   3633    3710    3725    3771    4789    4818    5048    5120    5404    5486    8314    8726
$RAND   042240           5535    8210#
$RDCHR  044264           8742#   8994
$RDDEC= ****** U         8995
$RDLIN= ****** U         8995
$RDOCT= ****** U         8995
$RDSZ = 000000           8763#
$REGAD  001160           1535#
$REG0   001162           1537#   6545    9469
$REG1   001164           1538#   9469
$REG10  001202           1545#   5712*   5713*   5714    5745*   5748    5897*   5898*   5899    5932*   5935    6770*   6800*
                         6839*   9473    9481
$REG11  001204           1546#   5746*   5933*   6771*   6801*   6840*   9473    9481
$REG12  001206           1547#   5747*   5934*   6772*   6802*   6833*   6835*   9473    9481
$REG13  001210           1548#   5749*   5751*   5753*   5754*   5936*   5938*   5940*   5941*   9473    9481
$REG14  001212           1549#   5748*   5750*   5752*   5935*   5937*   5939*   9476    9481
$REG15  001214           1550#   5755*   5756*   5942*   5943*   9476    9481
$REG16  001216           1551#   9477    9481
$REG17  001220           1552#   9477
$REG2   001166           1539#   9469
$REG20  001222           1553#   9477
$REG21  001224           1554#   9477
$REG22  001226           1555#   9477
$REG23  001230           1556#   9477
$REG24  001232           1557#   9477
$REG25  001234           1558#   9477
$REG26  001236           1559#   6412    6576
$REG27  001240           1560#
$REG3   001170           1540#   9469
$REG30  001242           1561#
$REG31  001244           1562#
$REG32  001246           1563#
$REG33  001250           1564#
$REG34  001252           1565#
$REG35  001254           1566#
$REG36  001256           1567#   6555*   9472
$REG37  001260           1568#   6556*   9472
$REG4   001172           1541#
$REG5   001174           1542#   5279    5727*   5735    5741*   5914*   5922    5928*   6360*   6361*   6552    6934*   6935*
                         6972    7044*   7066*   7078*   7092    7095    9473    9480
$REG6   001176           1543#   5302    5305    5728*   5737    5742*   5915*   5924    5929*   7045*   7048*   7061    7063*
                         7065*   7085*   7086*   7087*   7088*   7091*   7093    7094    9473    9480
$REG7   001200           1544#   5287    5729*   5739    5743*   5916*   5926    5930*   7046*   7047*   7051    7054*   7058*
                         7060*   7092*   7094*   7096*   9473    9480
$RESRE  045250           8944#   8996
$R2A  = ****** U         8997
$SAVRE  045212           8928#   8995
$SCOPE  044574           3524    8823#
$SETUP= 000037           3501#   3523    3524    3526    3528    3530    3532    3533    3535    4427    8689    8714    8721
                         8731    8769    8824
```

```
$STUP = 177777           3501#
$SUPRS  043356           4359    4386    5443    5470    8517#
$SVLAD  044674           8833    8839    8845#
$SVPC = 000230           1466#   1471
$SWR  = 163000           1316#   1318#   1323    1324    1325    1326    1327    1328    1329    1583    1584    3533    3535
                         3536    3867    3890    3938    3993    4038    4072    4144    4221    4318    4422    4428    4438
                         4444    4445    8680    8681    8682    8683    8684    8692    8699    8711    8714    8726    8817
                         8818    8819    8820    8824    8836    8838    8839    8840    8845    8848    8851    8854
$SWREG  001352           1605#   3556
$SWRMK= 000000           8820
$TESTN  001334           1596#   3867*   3890*   3938*   3993*   4038*   4072*   4144*   4221*   4318*   8846*
$TKB    001146           1528#   4730    4772*   8729    8746    8752
$TKS    001144           1527#   4741*   4774*   8729    8744    8750
$TMP0   001262           1569#
$TMP1   001264           1570#   6225*   6226*   6234
$TMP10  001302           1577#   6238*   6239*   6240
$TMP11  001304           1578#   4258*   4260*   4264    4278*   4280*   4284    4348*   4363*   4378*   4390*   4575*   4580*
                         4585*   4594    4606*   4607*   4657*   4662    4669*   4670*   5204*   5210
$TMP12  001306           1579#   4264*   4265*   4266    4268*   4269    4284*   4285*   4286    4288*   4289    4328*   4332
                         4370*   4578*   4588*   4611*   4658*   4671*   5205*   5211
$TMP13  001310           1580#   4346*   4361*   4362    4576*   4581*   4586*   4597    4609*   4610*   5206*   5212
$TMP14  001312           1581#
$TMP15  001314           1582#
$TMP2   001266           1571#   6221*   6249
$TMP3   001270           1572#   6222*   6250
$TMP4   001272           1573#   6223*   6251
$TMP5   001274           1574#
$TMP6   001276           1575#
$TMP7   001300           1576#
$TN   = 000012           1306#   1316    3854    3867#   3877    3890#   3928    3938#   3973    3993#   4024    4038#   4047
                         4072#   4124    4144#   4193    4221#   4297    4318#
$TNPWR  043272           8465    8466    8486#
$TPB    001152           1530#   8303*   8314
$TPFLG  001157           1534#   8252    8314
$TPS    001150           1529#   8301    8314
$TRAP   045306           3528    8964#
$TRAP2  045330           8975#   8986
$TRP  = 000012           8979#   8988#   8989#   8990#   8991#   8992#   8994    8995#   8996#   8997#   8998#
$TRPAD  045342           8969    8986#
$TSTM   001004           1491#
$TSTNM  001102           1507#   3564*   3613*   3714*   3779*   4005*   4402*   4427*   5803*   6400    8691    8726    8816
                         8845*   8846    8851    8854
$TYPBN= ****** U         8992
$TYPDS  043644           8619#   8991
$TYPE   042342           8252#   8885    8979    8987
$TYPEC  042554           8282    8289    8296    8301#   8302
$TYPEX  042622           8307    8309    8312#
$TYPOC  043442           8560#   8988
$TYPON  043456           8559    8562#   8990
$TYPOS  043416           8555#   8989
$UNIT   001342           1599#
$UNITM  001010           1493#
$USWR   001354           1606#
$VECT1  001400           1631#   3590
$VECT2  001402           1632#
$XTSTR  044604           8827#
```

```
$$GET4= 000000           4438#
$.$.SM= ****** U         2866    7171    7406    7597    7608    7646    7960    8001    8013    8067    8089    8107    8134
                         8143
$OFILL  043641           8556*   8560*   8570    8605#
$40CAT= ****** U         8701    8824
.       = 055151         1445#   1449#   1454#   1456#   1466    1467#   1469#   1471#   1472#   1478    1479#   1481#   1483#
                         1504#   1588    2952#   2953#   2958#   2974#   2981#   2982#   2984#   2989#   3079#   3080#   3081#
                         3082#   3083#   3084#   3085#   3086#   3087#   3088#   3089#   3090#   3091#   3092#   3093#   3094#
                         3140#   3143#   3146#   3151#   3154#   3157#   3162#   3165#   3168#   3173#   3176#   3179#   3215#
                         3218#   3221#   3226#   3229#   3232#   3521    3535    3536    4445    8314    8386    8444#   8506#
                         8673#   8726    8729    8769    8777    8794#   8854    8906#   9748#
.$ASTA= ****** U         8858    8861
.$X     = 001000         1478#   1483
..HIGH  000226           1460#
..LOW   000224           1459#
```

```
CKEXIT  2166#   6306
COMMEN     1#   1442#
ENDCOM     1#   1442#
ERROR   1336#   3832    4934    4953    5726    5744    5757    5913    5931    5944    5989    6362    6492    6496    6503
        6507    6511    6515    6519    6523    6527    6531    6533    6634    6639    6646    6648    6654    6656    6665
        6667    6669    6673    6677    6682    6687    6691    6695    6698    6702    6706    6710    6714    6718    6722
        6727    6732    6737    6742    6747    6752    6762    6779    6782    6792    6805    6807    6824    6841    6850
        6854    6859    6864    6869    6903    6905    6936
ESCAPE     1#   1442#
GETPRI     1#   1442#
GETSWR     1#   1442#
MSG     3854#   3856    3877#   3879    3928#   3930    3973#   3975    4024#   4026    4047#   4049    4124#   4126    4193#
        4195    4297#   4299
MULT       1#   1442#
NEWTST     1#   1442#   3854    3877    3928    3973    4024    4047    4124    4193    4297
NEXTAD  2162#
NXTAD   2158#
PARMBK     1#   2750    2782
PARMDF     1#   2656
POP        1#   1442#   8229    8660    8900    8901    8949
PUSH       1#   1442#   8210    8619    8861    8863    8884    8929
P.DRVR     1#   2814
REGDEF     1#   2438
REPORT     1#   1442#
R.DRIV     1#   7100
R.QDRV     1#
SCOPE   1337#   3866    3889    3937    3992    4037    4071    4143    4220    4317    4426
SETPRI     1#   1442#
SETTRA  8979#   8988    8989    8990    8991    8994    8995    8996    8997
SETUP      1#   1442#   3516
SKIP       1#   1442#
SLASH      1#   1442#
SPACE   1442#
STARS      1#   1295    1302    1442#   1464    1475    1477    1484    1500    1588    1591    3606    3610    3854    3865
        3877    3888    3928    3936    3973    3991    4024    4036    4047    4070    4124    4142    4193    4219    4297
        4316    4420    4448    4456    4525    4529    4566    4571    4618    4624    4678    4683    4705    4710    4725
        4727    4762    4770    4779    4786    4797    4802    4832    4838    4854    4862    4881    4886    4901    4914
        5002    5006    5021    5028    5099    5103    5138    5145    5194    5199    5220    5228    5255    5267    5317
        5338    5415    5419    5450    5454    5523    5529    5553    5556    5575    5584    5622    5631    5669    5677
        5775    5784    5824    5829    5848    5853    5874    5881    5963    5966    6019    6024    6068    6071    6093
        6097    6115    6121    6153    6156    6177    6185    6210    6220    6253    6259    6287    6303    6320    6322
        6346    6352    6366    6375    6475    6485    6536    6594    6602    6964    6968    7007    7009    7041    7043
        7071    7076    7109    7140    7174    7214    7577    7594    7612    7636    7683    7696    7715    7741    7760
        7782    7840    7867    8145    8158    8201    8237    8317    8327    8345    8356    8408    8447    8509    8532
        8609    8676    8728    8734    8798    8801    8813    8856    8913    8958
SWRSU      1#   1292#   1442#   3537#
S.DRVE     1#   2590
TEST    2154#   3854    3877    3928    3973    4024    4047    4124    4193    4297
TRMTRP  8979#
TYPBIN     1#   1442#
TYPDEC     1#   1442#
TYPNAM     1#   1442#
TYPNUM     1#   1442#
TYPOCS     1#   1442#
TYPOCT     1#   1442#
TYPTXT     1#   1442#
```

```
$DECBN       1#
$OCTBN       1#    8143
$$CMRE    1498#    1537    1538    1539    1540    1541    1542    1543    1544    1545    1546    1547    1548    1549    1550
          1551     1552    1553    1554    1555    1556    1557    1558    1559    1560    1561    1562    1563    1564    1565
          1566     1567    1568
$$CMTM    1498#    1569    1570    1571    1572    1573    1574    1575    1576    1577    1578    1579    1580    1581    1582
$$ESCA       1#    1442#
$$NEWT       1#    1442#   3854    3877    3928    3973    4024    4047    4124    4193    4297
$$SET     8979#    8988    8989    8990    8991    8994    8995    8996    8997
$$SETM    3553#
$$SKIP       1#    1442#
.EQUAT       1#    1292#   1332
.HEADE       1#    1292#   1307
.KT11        1#
.SETUP       1#    1292#   3501
.SWRHI       1#    1292#   1319
.SWRLO    1292#    1330#
.$ACT1       1#    1292#   1462
.$APTB       1#    1589#
.$APTH       1#    1292#   1473
.$APTY       1#    1292#   8854
.$ASTA       1#
.$CATC       1#    1292#   1443
.$CMTA       1#    1292#   1498
.$DB2D       1#    1292#   8445
.$DB2O       1#    1292#   8406
.$DIV        1#
.$EOP        1#    1292#   4418
.$ERRO       1#    1292#   8674
.$ERRT       1#
.$MULT       1#
.$POWE       1#    1292#
.$RAND       1#    1292#   8190
.$RDDE       1#
.$RDOC       1#
.$READ       1#    1292#   8726
.$R2AZ       1#
.$SAVE       1#    1292#   8911
.$SB2D       1#
.$SB2O       1#
.$SCOP       1#    1292#   8811
.$SIZE       1#    1292#
.$SUPR       1#    1292#   8507
.$TRAP       1#    1292#   8956
.$TYPB       1#
.$TYPD       1#    1292#   8607
.$TYPE       1#    1292#   8235
.$TYPO       1#    1292#   8530
.$40CA       1#
.1170        1#

. ABS.  055151     000

 ERRORS DETECTED:  0
```

```
 DSKZ:CZR6QC.BIN,DSKZ:CZR6QC.LST/CRF/SOL/NL:TOC/EQ:QNEWSW=DSKZ:CZR6QC.SML,DRIV10.P11,CZR6QC.P11
 RUN-TIME: 71 105 8 SECONDS
 RUN-TIME RATIO: 314/186=1.6
 CORE USED:  54K  (107 PAGES)
```