RK611/06

PERFORMANCE EXERCISER

CZR6PC0

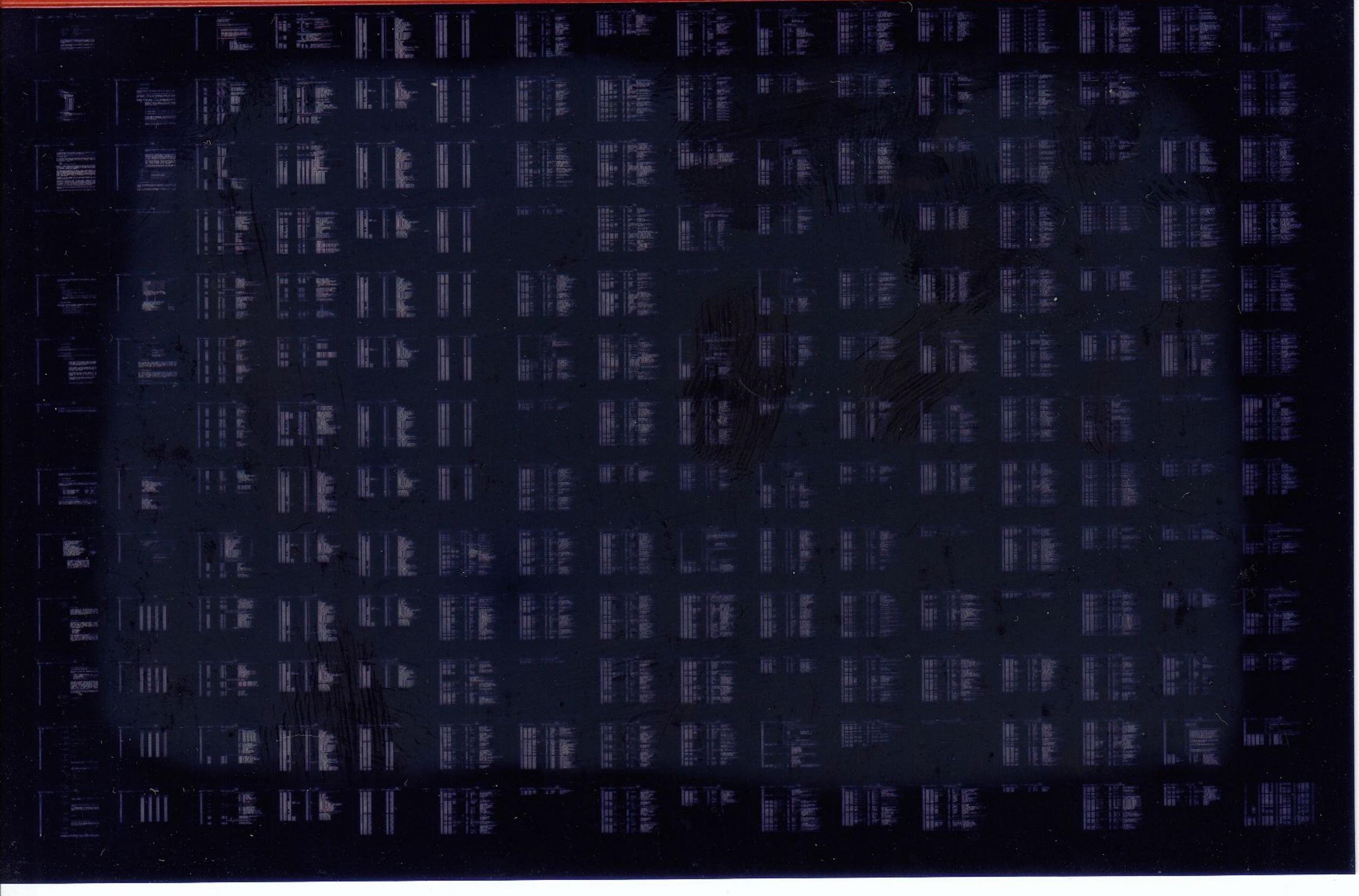
AH-9145C-MC

FICHE 1 OF 2

MAR 1978

digital

MADE IN USA



RK611/06

PERFORMANCE EXERCISER

CZR6PC0

AH-9145C-MC

FICHE 2 OF 2

MAR 1978

digital

MADE IN USA

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

IDENTIFICATION

PRODUCT CODE:	AC-9144C-MC
PRODUCT NAME:	CZR6PC0 RK611/06 PERF EXERCISER
DATE:	FEB 1978
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	ROY SPITZER

TABLE OF CONTENTS

1.0 ABSTRACT

THE RK06-RK07 PERFORMANCE EXERCISER PROGRAM WILL EXERCISE IN A RANDOM OVERLAPPED MANNER 1 TO 8 RK06-RK07 DISK DRIVES ATTACHED TO THE SAME RK06-RK07 UNIBUS CONTROLLER IN A STAND ALONE MODE.

DRIVES UNDER TEST CAN BE ADDED TO OR DROPPED FROM THE TESTING SEQUENCE BY OPERATOR COMMAND.

AT ANY GIVEN POINT IN TIME, THE NEXT DRIVE ON WHICH AN OPERATION IS TO BE INITIATED IS CHOSEN RANDOMLY FROM AMONG THE RK06 DRIVES UNDER TEST NOT HAVING ANY ERROR CONDITIONS PRESENT AND NOT CURRENTLY UNDERGOING ERROR RECOVERY. THE COMMAND IS THEN CHOSEN RANDOMLY FROM THE FOLLOWING SET OF COMMANDS:

READ
WRITE
WRITE FOLLOWED BY WRITE CHECK

THEN THE CYLINDER, TRACK, SECTOR, WORD COUNT, AND DATA ARE DETERMINED RANDOMLY. THE DATA WRITTEN IS RANDOMLY SELECTED FROM A SET OF FIXED DATA PATTERNS. TO OPTIMIZE THE THROUGHPUT ON THE RK06 CONTROLLER EACH DATA TRANSFER COMMAND IS TRANSLATED TO AN EXPLICIT SEEK COMMAND FOLLOWED BY THE DATA TRANSFER COMMAND.

DRIVE ERRORS ARE QUEUED UP AS THEY OCCUR. WHILE ERROR RECOVERY IS BEING PROCESSED ON ONE DRIVE ALL OTHER DRIVES HAVING NO ERRORS WILL BE EXECUTING ORDERS GENERATED RANDOMLY AS DESCRIBED ABOVE. EACH DRIVE ERROR IS PROCESSED ON A FIRST-IN-FIRST-OUT BASIS. THIS MEANS THAT A DRIVE ERROR WILL BE REPORTED AND ERROR RECOVERY COMPLETELY PROCESSED BEFORE THE NEXT DRIVE ERROR IS REPORTED AND ERROR PROCESSING HAS BEGUN. ERROR RECOVERY IS PROCESSED AS DESCRIBED IN THE RK06 DISK DRIVE SPECIFICATION.

REPORTING OF SYSTEM ERRORS SUCH AS CONTROLLER PROBLEMS WILL BE IMMEDIATE UPON OCCURRENCE, NOT DEFFERRED, PERHAPS CAUSING PREMATURE PROGRAM TERMINATION.

PERFORMANCE STATISTICS ARE KEPT ON EACH DRIVE. THESE STATISTICS INCLUDE BOTH OPERATION COUNTS AND ERROR SUMMARY INFORMATION. (SEE SECTION 6.1) THESE STATISTICS WILL BE INITIALIZED WHEN TESTING BEGINS ON A DRIVE. AT ANY TIME AFTER TESTING BEGINS ON THE DRIVE(S) UNDER PERFORMANCE EVALUATION, THE OPERATOR CAN DEMAND THESE PERFORMANCE STATISTICS. IF A REAL TIME CLOCK IS AVAILIBLE, THE OPERATOR HAS THE OPTION OF HAVING THE PROGRAM GIVE PERIODIC PERFORMANCE SUMMARIES.

ONCE A PARTICULAR DRIVE HAS AN UNRECOVERABLE ERROR (OTHER THAN A DATA TRANSFER AND SEEK INCOMPLETE), THAT DRIVE WILL BE AUTOMATICLY DEASSIGNED.

ONCE A PARATICULAR DRIVE HAS EXCEEDED AN ERROR THRESHOLD (DATA TRANSFER OR SEEK) SPECIFIED FOR A DRIVE, THAT DRIVE WILL BE

149 AUTOMATICLY DEASSIGNED UNLESS SWITCH 4 IS SET.

ONCE A PARTICULAR DRIVE HAS EXCEEDED THE MAXIMUM NUMBER OF PERMITTED OPERATIONS FOR THAT DRIVE, THE DRIVE WILL BE AUTOMATICALLY DESELECTED UNLESS SWITCH 5 IS SET.

2.0 REQUIREMENTS

2.1 EQUIPMENT

- PDP-11 WITH AT LEAST 16K OF MEMORY
- CONSOLE TERMINAL
- DECTAPE, PAPER TAPE READER; OR DECDISK
- RK611 CONTROLLER
- 1 TO 8 RK06-RK07 DISK DRIVE WITH FORMATTED PACK (16 BIT FORMAT)
- REAL TIME CLOCK (KW11-P OR KW11-L OPTIONAL) FOR INTERVAL PERFORMANCE REPORTING AND TIME ASSOCIATIONS WITH ERROR REPORTS

2.2 PRELIMINARY PROGRAMS

THE RK06-RK07 SUBSYSTEM IS ASSUMED TO BE BASICALLY OPERATIONAL AND FREE OF HARD FAULTS. THE FOLLOWING RK06 PROGRAMS ARE ASSUMED TO BE RUNNING WITHOUT ERROR BEFORE THIS PROGRAM IS RUN:

- RK611 DISKLESS CONTROLLER DIAGNOSTICS
- RK06-RK07 DRIVE DIAGNOSTICS
- RK611 FUNCTIONAL CONTROLLER DIAGNOSTIC
- RK06-RK07 SUBSYSTEM VERIFICATION
- RK06-RK07 PACK FORMATTER (IF PACKS ARE UNFORMATTED)

3.0 OPERATING PROCEDURE

3.1 LOADING PROCEDURE

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

3.2 STARTING PROCEDURES

AFTER THE PROGRAM IS LOADED, THE OPERATOR STARTS THE PROGRAM AT ONE OF THE FOLLOWING LOCATIONS:

LOCATION	FUNCTION
200	INITIALIZE ALL DRIVE STATISTIC TABLES, THE RANDOM NUMBER SEED, AND USE DEFAULT SYSTEM PARAMETERS. THE PROGRAM WILL THEN DO AUTO-SIZING OF ALL DRIVES ON THE BUS & BEGIN BY FIRST WRITING THOSE DRIVES & THEN BEGIN PERFORMANCE TESTING.
204	RESTART PROGRAM. IF DRIVE IS UNDER TEST, CONTINUE TESTING DRIVE AND DO NOT DESTROY STATISTICS. IF PACK IS BEING WRITTEN, START WRITING PACK FROM THE BEGINNING. THE PROGRAM WILL THEN TYPE READY TO BEGIN PERFORMANCE TESTING.
214	START PROGRAM AND ALTER SYSTEM PARMETERS. THE PROGRAM WILL THEN IDENTIFY ITSELF AND ASK FOR THE PARAMETERS AS DESCRIBED IN SECTION 3.3.2. THE PROGRAM WILL THEN TYPE READY TO BEGIN PERFORMANCE TESTING.
240	INITIALIZE ALL DRIVE STATISTIC TABLES, THE RANDOM NUMBER SEED, AND USE DEFAULT SYSTEM PARAMETERS. THE PROGRAM WILL THEN TYPE READY TO BEGIN PERFORMANCE TESTING.

NOTE: IF A POWER FAIL OCCURS, THE PROGRAM WILL AUTOMATICALLY PERFORM THE RESTART FUNCTION.

```
257
258
259                     THE PROGRAM IS NOW IN AN IDLE LOOP WAITING FOR THE INITIATION OF  RK06
260                     DRIVE TESTING.
261
262                     THE OPERATOR THEN STARTS PERFORMANCE EVALUATION OF THE  RK06  DRIVE(S)
263                     BY  TYPING  A  CONTROL-C  <C>  FOLLOWED  BY A TN OR PN AS DESCRIBED IN
264                     SECTION 4.1.  IF THE PACK DOES NOT HAVE  THE  PROPER  RANDOMLY  CHOSEN
```

PREDETERIMED TEST PATTERNS, THE OPERATOR MUST USE THE WN COMMAND AS DESCRIBED IN SECTION 4.1 BEFORE INITIATING THE PERFORMANCE EVALUATION SEQUENCE ON THAT DRIVE.

3.3 SYSTEM PARAMETERS

3.3.1 DIRECT MEMORY ALTERATION

AS PART OF THE PROGRAM START-UP PROCEDURE, THE FOLLOWING SYSTEM PARMETERS WILL BE LOADED BY THE OPERATOR OR DEFAULTED BY THE PROGRAM. THE OPERATOR MUST PHYSICALLY ALTER THE APPROIATE MEMORY LOCATIONS TO CHANGE THESE VALUES.

PARAMETER	TAG	DEFAULT VALUE
- KW11-P STATUS REGISTER ADDRESS	$LKCSR	172540
- KW11-P COUNTER BUFFER ADDRESS	$LKCSB	172542
- KW11-P VECTOR ADDRESS	$LPVEC	104
- KW11-L STATUS REGISTER ADDRESS	$LKS	177506
- KW11-L VECTOR ADDRESS	$LLVEC	100
- SYSTEM POWER (HERTZ)	HZ	60(DECIMAL)

3.3.2 PROGRAM MODIFIED PARAMETERS

THE FOLLOWING PARAMETERS ARE MODIFIED BY A 214 PROGRAM START.

A <CR> WILL DEFAULT CURRENT PARAMETERS. A CONTROL Z <↑Z> <CR> WILL DEFAULT ALL THE REST OF THE PARAMETERS. A CONTROL C <↑C> <CR> WILL RETURN TO FIRST PARAMETER ENTRY REQUEST FOR THIS DRIVE. A RUBOUT CANCELS THE LAST CHARACTER TYPED. A CONTROL U <↑U> CANCELS LINE BEING TYPED. ALL ENTRIES MUST BE TERMINATED WITH CARRIAGE RETURN.

PARMETER	DEFAULT VALUE
- CPU IDENTIFIER (OCTAL NUMBER 0-377 FOR DUAL ACCESS PORT DISCRIMINATION)	0
- OVERLAY OF LOADER BY PROGRAM? (ALLOWS GREATER EXPANSION OF MEMORY FOR BUFFER SPACE IF LOADER IS OVERLAYED)	NO
- MAXIMUM DATA BUFFER (SEE NOTE) UP TO 5888 (23 SECTORS)	SEE NOTE
- NUMBER OF DATA WORDS COMPARED PER READ (0 - MAXIMUM BUFFER LENGTH) EACH READ COMMAND ISSUED BY THE PROGRAM HAS AN IMPLICIT DATA COMPARISON ASSOCIATED WITH IT VERIFYING THAT THE DATA READ IS ONE OF LEGAL RANDOM PATTERNS.	3 WORDS
- INTERVAL BETWEEN AUTOMATIC PERFORMANCE TYPE-OUTS (EVERY 1-255 MINUTES OR NO AUTOMATIC PERFORMANCE TYPE-OUTS)	NO AUTOMATIC PERFORMANCE TYPE-OUTS
- RK611/RK06-RK07 UNIBUS ADDRESS	177440
- RK611/RK06-RK07 VECTOR ADDRESS	210
- RK611/RK06-RK07 PRIORITY	5

NOTE

THE PROGRAM WILL SELECT A MAXIMUM BUFFER SIZE WHICH IS DETERMINED BY THE MEMORY AVAILABLE. THE MAXIMUM BUFFER SIZE ASSIGNED BY THE PROGRAM WILL BE 5888 WORDS. THE DEFAULT VALUE OF THIS PARAMETER WEIGHTS THE OPERATIONS TOWARDS MAXIMIZING DATA TRANSFERS PER UNIT OF TIME. BY CHANGING THIS PARAMETER TO 1 SECTOR (256 WORDS), THE OPERATIONS WILL BE WEIGHTED TOWARDS MAXIMIZING SEEKS PER UNIT OF TIME.

3.4 SWITCHES

3.4.1 SWITCH 15

SW<15> = 1 HALT ON ERROR

IF THIS SWITCH IS SET, THE PROGRAM WILL NOT ISSUE ANOTHER COMMAND TO THE RK06 SUBSYSTEM WHEN AN ERROR OCCURS AND THE PROGRAM WILL HALT IN THE SYSMAC ERROR HANDLER. IF THE OPERATOR DESIRES TO CONTINUE FROM WHERE THE PROGRAM LEFT OFF, THE OPERATOR SIMPLY PRESSES THE CONTINUE SWITCH ON THE CONSOLE PANEL AND THE PROGRAM CONTINUES WITH THE NEXT RANDOMLY GENERATED COMMAND.

3.4.2 SWITCH 14

SW<14> = 0 NORMAL OPERATION

THE OPERATION THE PROGRAM IN REGARD TO DRIVE SELECTION AND COMMAND GENERATION IS NORMAL. NORMAL OPERATION IS TO RANDOMLY SELECT A DRIVE, STORE THE PREVIOUS OPERATION EXECUTED ON THAT DRIVE, AND GENERATE A NEW RANDOM OPERATION TO BE EXECUTED ON THAT DRIVE. THE GENERATED OPERATION INCLUDES RANDOMLY GENERATED VALUES FOR THE FOLLOWING PARAMETERS:

- DRIVE COMMAND
- CYLINDER ADDRESS
- TRACK ADDRESS
- SECTOR ADDRESS
- WORD COUNT
- DATA PATTERN

THESE VALUES ARE STORED AS THE PREVIOUS OPERATION WHEN A NEW OPERATION IS GENERATED.
IT IS IMPORTANT TO NOT THAT AT ANY POINT IN TIME THERE IS BOTH A CURRENT OPERATION AND A STORED PREVIOUS OPERATION ASSOCIATED WITH EACH DRIVE UNDER TEST. THE CURRENT OPERATION MAY HAVE ALREADY BEEN EXECUTED AND A NEW COMMAND HAS NOT YET BEEN GENERATED OR THE COMMAND IS WAITING TO BE EXECUTED.

SW<14> = 1 LOOP ON CURRENT OPERATIONS (TEST)

DO NOT GENERATE NEW RANDOM OPERATION BUT CONTINUE TO EXECUTE CURRENT OPERATIONS.

WHEN SWITCH 14 IS SET THE GENERATION OF A NEW OPERATION IS INHIBITED AND THE PROGRAM OPERATES AS FOLLOWS:

1.) A DRIVE IS RANDOMLY SELECTED.
2.) A SEEK IS EXECUTED ON THE DRIVE TO THE CYLINDER ADDDRESS STORED IN THAT DRIVE'S PREVIOUS OPERATION.
3.) THE CURRENT OPERATION FOR THAT DRIVE IS EXECUTED.
4.) REPEAT THE ENTIRE PROCESS.

USEAGE THIS SWITCH CAN BE USED IN CONJUNCTION WITH SWITCH 15 TO ENABLE THE REPETITION OF A RANDOMLY GENERATED OPERATION. THE OPERATOR WOULD START WITH SWITCH 15 SET AND SWITCH 14 RESET. WHEN THE PROGRAM HAS INDICATED THAT AN ERROR HAS OCCURRED BY HALTING. THE CURRENT OPERATION FOR THE DRIVE IN ERROR WILL NOT HAVE BEEN MODIFIED WHEN THE PROGRAM HALTS. THE OPERATOR CAN THEN CAUSE ALL DRIVES TO LOOP ON THEIR INDIVIDUAL CURRENT OPERATIONS BY SETTING SWITCH 14, RESETTING SWITCH 15, AND HITTING CONTINUE.

TO HELP ISOLATE PROBLEMS, THE OPERATOR CAN DROP THE RK06 DRIVE(S) FROM THE TEST SEQUENCE AT ANY TIME BY USING THE DN COMMAND AS DESCRIBED IN SECTION 4.1.

3.4.3 SWITCH 13

SW<13> = 1 INHIBIT ERROR TYPE OUT

3.4.4 SWITCH 12

NO EFFECT.

3.4.5 SWITCH 11

NO EFFECT.

3.4.6 SWITCH 10

SW<10> = 1 RING TELETYPE BELL IF ERROR

3.4.7 SWITCH 9

NO EFFECT.

3.4.8 SWITCH 8

NO EFFECT.

3.4.9 SWITCH 7

NO EFFECT.

3.4.10 SWITCH 6

SW<6> = 1 INHIBIT AUTOMATIC DESELECT FOR CLEARABLE UNSAFES, SPEED LOSS, AND AC LOW.

3.4.11 SWITCH 5

SW<5> = 1 INHIBIT AUTOMATIC DRIVE DEASSIGNMENT IF OPERATION COUNT THRESHOLD IS EXCEEDED.

3.4.12 SWITCH 4

SW<4> = 1 INHIBIT AUTOMATIC DRIVE DEASSIGNMENT IF ERROR THRESHOLD EXCEEDED.

3.4.13 SWITCH 3

SW<3> = 1 DISPLAY ENTIRE SECTOR READ BEFORE STARTING RETRY SEQUENCE

ALL HARDWARE DETECTED DATA ERRORS WILL BE RETRIED ON A SECTOR BASIS. IF THIS SWITCH IS SET, THE DATA SECTOR IN ERROR WILL BE DISPLAYED ENTIRELY BEFORE THE RETRY SEQUENCE BEGINS ON THE SECTOR IN ERROR.

3.4.14 SWITCH 2

NO EFFECT.

3.4.15 SWITCH 1

SW<1> = 1 INHIBIT SOFTWARE DATA COMPARISONS

THE IMPLICIT DATA COMPARISONS ASSOCIATED WITH EACH READ WILL NOT BE PERFORMED WHEN THIS SWITCH IS SET. THE ONLY DATA CHECKING PERFORMED WILL BE HARDWARE DATA CHECKING.

3.4.16 SWITCH 0

NO EFFECT.

3.5 RUN TIME

THIS PROGRAM IS DESIGNED TO RUN FOR AN EXTENDED PERIOD OF TIME. (ESTIMATED RUN TIME WITH PRESENT DEFAULT PARAMETERS IS APPROXIMATELY 250 HOURS PER DRIVE.) MEANINGFUL INFORMATION CAN STILL BE DERIVED BY

```
563                     RUNNING THIS PROGRAM FOR AN HOUR OR LESS SINCE STATISTICS ARE TAKEN AS
564                     THE PROGRAM IS RUNNING.
```

4.0 OPERATING PROCEDURE

4.1 OPERATOR COMMANDS

AFTER THE PROGRAM IS LOADED AND HAS PRINTED OUT "READY TO BEGIN PERFORMANCE TESTING", THE OPERATOR CAN ONLY TYPE A CONTROL-G <G> OR A CONTROL-C <C>.

A CONTROL-G <G> IS USED FOR SOFTWARE SWITCH REGISTER MODIFICATION.

UPON RECEIVING A CONTROL-C <C> THE PROGRAM WILL RESPOND WITH "PLEASE TYPE COMMAND". THE FOLLOWING TWO CHARACTER COMMANDS FOLLOWED BY CARRIAGE RETURN WILL BE THE ONLY COMMANDS ACCEPTED: TN, PN, DN, SN, WN.

THE SECOND CHARACTER N=0,1,...,7 OR A DESIGNATES WHICH DRIVE IS REFERENCED BY THE COMMAND. IF N=A, ALL ADDRESSABLE DRIVES ARE REFERENCED. THE DESCRIPTION OF EACH OPERATOR COMMAND FOLLOWS:

- TN INITIATE TESTING OF DRIVE N AND USE PREVIOUS PARAMETERS. (DEFAULT PARMETERS ARE USED IF PN COMMAND HAS NOT PREVIOUSLY BEEN ISSUED TO THE DRIVE(S) AT THE BEGINNING OF THEIR TEST SEQUENCE.) IF N=A, INITIATE TESTING ON ALL DRIVES.

- PN THIS COMMAND HAS TWO USES:

  1.) CHANGE CURRENT DRIVE PARAMETERS AND INITIATE TESTING ON DRIVE N.

  2.) CHANGE CURRENT DRIVE PARAMETERS AND CONTINUE TESTING ON DRIVE N.

  IN BOTH CASES THE PROGRAM WILL THEN ASK FOR PARMAETERS IN THE ORDER AS DESCRIBED IN SECTION 5.5. A <CR> INDICATES DEFAULT THIS PARAMETER AND A CONTROL-Z <Z> INDICATES THE THE REST OF THE PARAMETERS FOR THIS DRIVE ARE TO BE DEFAULTED.

  IF N=A, PERFORM PN COMMAND ON ALL DRIVES.

- DN DROP DRIVE FROM TESTING SEQUENCE AND PRINT PERFORMANCE SUMMARY. IF N=A, DROP ALL CURRENTLY BEING TESTED DRIVES FROM TESTING SEQUENCE

- SN DEMAND PERFORMANCE SUMMARY ON DRIVE N. IF N=A DEMAND PERFORMANCE SUMMARY ON ALL CURRENTLY BEING TESTED DRIVES.

- WN WRITE AND VERIFY THE RK06 DISK PACK ON DRIVE N WITH A RANDOM SET OF LEGAL PATTERNS. THIS IS USED PRIOR TO THE START OF THE TEST SEQUENCE TO GUARENTEE THAT THE PROPER PREDETERMINED TEST PATTERNS HAVE BEEN WRITTEN ON THE PACK. (SEE APPENDIX A FOR DETAILS) AFTER THE PACK HAS BEEN COMPLETELY WRITTEN TESTING WILL BEGIN USING THE PREVIOUS PARAMETERS AS DESCRIBED IN THE TN COMMAND.

  WRITING OF THE PACKS WILL TAKE 4 TO 5 MINUTES PER PACK WITH MAXIMUM BUFFER SIZE. THE WRITING OF THE PACK MAY TAKE LONGER IF THE MAXIMUM BUFFER SIZE SPECIFIED IS SMALL. RANDOM EXERCISE CAN BE DONE ON SOME DRIVES WHILE EXERCISE ON OTHER DRIVES ARE IN PROGRESS.

NOTE: IF AN ERROR OCCURS DURING THE DRIVE ASSIGNMENT SEQUENCE OR WHILE WRITING THE RANDOM SET OF LEGAL PATTERNS, THE ERROR WILL BE REPORTED AND TESTING WILL NOT START.

EXAMPLES:

COMMAND SEQUENCE	ACTION TAKEN
TA<CR>	INITIATE TESTING ON ALL DRIVES ON SYSTEM
T1<CR>	INITIATE TESTING ON DRIVE 1
T3<CR>	INITIATE TESTING ON DRIVE 3 (TESTING STILL CONTINUES ON DRIVE 1)
SA<CR>	GATHER STATISTICS ON ALL DRIVES UNDER TEST.
DA<CR>	DROP ALL DRIVES UNDER TEST.

4.2 DRIVE PARAMETERS (PER EACH DRIVE)

A SET OF THESE PARAMETERS EXISTS FOR EACH DRIVE. A TN COMMAND WILL UTILIZE THE PREVIOUSLY ESTABLISHED PARAMETERS FOR THE DRIVE(S). A PN COMMAND WILL ALLOW THE OPERATOR TO ALTER ALL THE PARAMETERS FOR THE DRIVE(S).

A <CR> WILL DEFAULT CURRENT PARAMETERS. A CONTROL Z <↑Z> <CR> WILL DEFAULT ALL THE REST OF THE PARAMETERS. A CONTROL C <↑C> <CR> WILL RETURN TO FIRST PARAMETER ENTRY REQUEST. FOR THIS DRIVE. A RUBOUT

```
677                     CANCELS THE LAST CHARACTER TYPED.  A CONTROL U <↑U> CANCELS LINE BEING
678                     TYPED.  ALL ENTRIES MUST BE TERMINATED WITH CARRIAGE RETURN.
```

	PARAMETER	DEFAULT VALUE (RK06/RK07)
-	MAXIMUM CYLINDER FOR START OF DATA TRANSFER 0 - 632/1456	632/1456
-	MINIMUM CYLINDER FOR START OF DATA TRANSFER 0 - 632/1456	0
-	MAXIMUM TRACK FOR START OF DATA TRANSFER 0 - 2	2
-	MINIMUM TRACK FOR START OF DATA TRANSFER 0 - 2	0
-	MAXIMUM SECTOR FOR START OF DATA TRANSFER 0 - 25	25
-	MINIMUM SECTOR FOR START OF DATA TRANSFER 0 - 25	0
-	RATIO OF READ/WRITE EVERY COMMAND RANDOMLY GENERATED BY THE PROGRAM WILL BE EITHER A READ OR A WRITE. FOR EACH DRIVE ONE CAN WEIGHT THE COMMANDS TOWARDS READING OR WRITING. THE POSSIBLE WEIGHTING RATIO'S ARE: READ ONLY =0 7/1 =1 3/1 =2 5/3 =3 1/1 =4 3/5 =5 1/3 =6 1/7 =7	5/3
-	DO WRITE CHECK AFTER EVERY WRITE COMMAND OTHERWISE, WRITE CHECK COMMANDS WILL BE ISSUED RANDOMLY AFTER WRITE COMMANDS.	NO
-	CORRECTABLE READ ERROR THRESHOLD	10
-	UNCORRECTABLE READ ERROR THRESHOLD	5
-	SEEK ERROR THRESHOLD	5
-	INHIBIT ERROR CORRECTION	NO
-	OPERATION COUNT THRESHOLD*65K	4.290*10**9
-	DATA TRANSFER THRESHOLD*520K WORDS	291*10**12
-	SAMPLED COMPARES	YES
-	UP TO 5 EXCLUDED PACK AREAS (NO DATA TRANSFER WILL BE DONE TO THESE AREAS.)	NO EXCLUDED PACK AREAS

4.2.1 FORMATS FOR PACK AREA EXCLUSION

FORMAT	ACTION
CYL<CR>	ALL TRACK ON CYLINDER SPECIFIED WILL BE EXCLUDED.
CYL,TRK<CR>	THE ENTIRE TRACK WILL BE EXCLUDED.
CYL1,TRK1,CYL2,TRK2<CR>	ALL TRACKS FROM CYLINDER 1 TRACK 1 TO AND INCLUDING CYLINDER 2 TRACK 2 WILL BE EXCLUDED.

4.3 DIAGNOSTIC DUMP

DUE TO THE COMPLEXITY OF THIS PROGRAM A DUMP OF THE PROGRAM TABLES MUST ACCOMPANY EACH PROBLEM REPORT. THIS DUMP IS PROVIDED IN THE PROGRAM AND IS ACTIVATED BY STARTING THE PROGRAM AT LOCATION 220. THIS DUMP WILL PRINT THE PROCESSOR REGISTERS WHEN THE DUMP IS CALLED. IT WILL THEN PRINT ABOUT 12-15 TELETYPE PAGES OF SOFTWARE TABES NECESSARY TO DETERMINE WHAT THE PROBRAM WAS DOING AT A GIVEN TIME. APPROXIMATE TIME OF DUMP IS 10-12 MINUTES.

5.0 PROGRAM DESCRIPTION

THIS PROGRAM ASSUMES THAT ONLY THE PATTERNS WRITTEN BY THIS PROGRAM WILL BE ON THE PACK. TO PRECONDITION THE PACK A WN COMMAND MUST BE DONE. PACK SERIAL, DRIVE SERIAL NUMBER, THE TIME, AND DRIVE N UNDER TEST WILL BE PRINTED AT THE END OF PACK PRECONDITIONING. THE OPERATION OF THE PROGRAM IS DEPENDENT ON THE SWITCH REGISTER, SYSTEM PARAMETERS, AND DRIVE PARAMETERS. DRIVES MAY BE DROPPED FROM TEST BY USING THE DN COMMAND. IF A DRIVE IS DROPPED FOR ANY REASON, PERFORMANCE STATISTICS WILL AUTOMATICALLY BE PRINTED.

5.1 PROGRAM RESTRICTIONS

THIS PROGRAM WILL NOT RUN IN CHAIN MODE UNDER XXDP. NO END-OF-PASS INDICATOR IS PROVIDED FOR ACT. THE APT PROTOCOL HAS NOT BEEN

```
   780                          DETERMINED YET.
```

6.0 PRINT OUTS

6.1 PERFORMANCE SUMMARY TYPEOUT

THE PERFORMANCE SUMMARY WILL CONTAIN THE FOLLOWING INFORMATION:

- DRIVE NUMBER
- DRIVE SERIAL NUMBER
- TIME OF REPORT (IF REAL TIME CLOCK IS AVAILIABLE)
- NUMBER OF ORDERS PERFORMED BY DRIVE
- NUMBER OF SEEK OPERATIONS PERFORMED
- TOTAL NUMBER OF WORDS WRITTEN BY DRIVE*65K
- TOTAL NUMBER OF WORDS READ BY DRIVE*65K
- NUMBER OF SOFT DATA ERRORS
  - (ECC CORRECTABLE)
  - (REREAD CORRECTABLE)
  - (OFFSET CORRECTABLE)
- NUMBER OF HARD DATA ERRORS
- NUMBER OF SEEK INCOMPLETES
- NUMBER OF MISPOSITIONING ERRORS
- TOTAL NUMBER OF ALL OTHER ERRORS.

6.2 ERROR REPORTS

THE ERROR REPORTS WILL CONTAIN THE FOLLOWING INFORMATION:

- DRIVE NUMBER
- DRIVE SERIVAL NUMBER
- TIME ERROR OCCURRED (IF REAL TIME CLOCK IS AVAILIABLE)
- DESCRIPTION OF ERROR
- PRESENT COMMAND
- PREVIOUS COMMAND
- PRESENT POSITION
- PREVIOUS POSITION
- ALL DRIVE STATUS REGISTERS
- ALL CONTROLLER REGISTERS
- MEMORY ADDRESS, GOOD DATA AND BAD DATA (IF DATA ERROR)
- OFFSET VALUE (IF APPLICABLE)
- HEADER READ (IF MISPOSITIONING ERROR)
- ECC PATTERN AND POSITION REGISTER IF DATA CHECKED

APPENDIX A

FIGURE A-1 SHOWS THE DATA EXPECTED BY THIS PROGRAM FOR EACH SECTOR OF THE RK06 PACK USED BY THIS PROGRAM.

CPU ID	CPU IDENTIFIER SUPPLED BY THE OPERATOR
PAT	PATTERN USED
PATC	PATTERN COUNT (NUMBERR OF WORDS OF PATTERN WRITTEN)

FIGURE A-1

EXPECTED DATA FORMAT

866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905

DATA PATTERNS USED

PATTERN 0	PATTERN 1	PATTERN 2	PATTERN 3
165555	026455	007417	052525
133333	151322	170360	125252
165555	026455	007417	052525
133333	026455	007417	052525
165555	151322	170360	125252
133333	151322	170360	125252
165555	026455	007417	052525
133333	026455	007417	052525
165555	026455	007417	052525
133333	151322	170360	125252
165555	151322	170360	125252
133333	151322	170360	125252
165555	026455	007417	052525
133333	026455	007417	052525
165555	026455	007417	052525
133333	026455	007417	052525
165555	151322	170360	125252
133333	151322	170360	125252
165555	151322	170360	125252
133333	151322	170360	125252
165555	026455	007417	052525
133333	026455	007417	052525
165555	026455	007417	052525
133333	151322	170360	125252
165555	151322	170360	125252
133333	151322	170360	125252
165555	026455	007417	052525
133333	026455	007417	052525
165555	151322	170360	125252
133333	151322	170360	125252
165555	026455	007417	052525
133333	151322	170360	125252

```
906
907
908
909                 PATTERN 4     PATTERN 5     PATTERN 6     PATTERN 7
910
911                 000000        000000        000001        000001
912                 010421        177777        000003        000002
913                 021042        000000        000007        000004
914                 031463        000000        000017        000010
915                 042104        177777        000037        000020
916                 052525        177777        000077        000040
917                 063146        000000        000177        000100
918                 073567        000000        000377        000200
919                 104210        000000        000777        000400
920                 114631        177777        001777        001000
921                 125252        177777        003777        002000
922                 135673        177777        007777        004000
923                 146314        000000        017777        010000
924                 156735        000000        037777        020000
925                 167356        000000        077777        040000
926                 177777        000000        177777        100000
927                 177777        177777        177776        100000
928                 167356        177777        177774        040000
929                 156735        177777        177770        020000
930                 146314        177777        177760        010000
931                 135673        000000        177740        004000
932                 125252        000000        177700        002000
933                 114631        000000        177600        001000
934                 104210        177777        177400        000400
935                 073567        177777        177000        000200
936                 063146        177777        176000        001000
937                 052525        000000        174000        000040
938                 042104        000000        170000        000020
939                 031463        177777        160000        000010
940                 021042        177777        140000        000004
941                 010421        000000        100000        000002
942                 000000        177777        000000        000001
```

```
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
```

PATTERN 10	PATTERN 11	PATTERN 12	PATTERN 13
177776	172666	153333	000000
177775	155555	066667	177777
177773	172666	153333	177777
177767	155555	066667	177777
177757	172666	153333	177777
177737	155555	066667	177777
177677	172666	153333	177777
177577	155555	066667	177777
177377	172666	153333	177777
176777	155555	066667	177777
175777	172666	153333	177777
173777	155555	066667	177777
167777	172666	153333	177777
157777	155555	066667	177777
137777	172666	153333	177777
077777	155555	066667	177777
077777	172666	153333	177777
137777	155555	066667	177777
157777	172666	153333	177777
167777	155555	066667	177777
173777	172666	153333	177777
175777	155555	066667	177777
176777	172666	153333	177777
177377	155555	066667	177777
177577	172666	153333	177777
177677	155555	066667	177777
177737	172666	153333	177777
177757	155555	066667	177777
177767	172666	153333	177777
177773	155555	066667	177777
177775	172666	153333	177777
177776	155555	066667	177777

PATTERN 14	PATTERN 15	PATTERN 16	PATTERN 17
177777	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
000000	172304	014561	133467
000000	172304	070627	133467
000000	172304	113431	133467
@			

```
                        ; *** REV 005 ***

                        .TITLE  CZR6PC0 RK611/06 PERF EXEC
                        ;*COPYRIGHT (C) 1976,1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ROY SPITZER
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
                        ;;**********************************************************************
                        ;*
                        ;*      THE RK06-RK07 PERFORMANCE EXERCISER WILL EXERCISE 1 TO 8
                        ;*      RK06-RK07 DISK DRIVES ATTACHED TO THE SAME RK06
                        ;*      UNIBUS CONTROLLER IN A DEDICATED STAND ALONE MODE.
                        ;*
                        ;*      UNDER NORMAL OPERATION SITUATIONS, THE DRIVE IS
                        ;*      CHOSEN RANDOMLY FROM ANY ONE OF THE RK06 DRIVES UNDER TEST
                        ;*      WHICH IS NOT CURRENTLY UNDERGOING ERROR RECOVERY.
                        ;*      TO OPTIMIZE THE THROUGHOUT ON THE RK06 CONTROLLER EACH
                        ;*      DATA TRANSFER COMMAND IS TRANSLATED INTO A SEEK FOLLOWED
                        ;*      BY THE DATA TRANSFER COMMAND.
                        ;*
                        ;;**********************************************************************
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        10            BELL ON ERROR
                        ;*         6            INHIBIT AUTOMATIC DRIVE DEASSIGNMENT IF UNSAFE,
                        ;*                      AC LOW, OR SPEED LOSS OCCURS
                        ;*         5            INHIBIT AUTOMATIC DRIVE DEASSIGNMENT WHEN
                        ;*                      MAXIMUM COMMAND COUNT FOR DRIVE IS EXCEEDED
                        ;*         4            INHIBIT AUTOMATIC DRIVE DEASSIGNMENT WHEN
                        ;*                      ERROR THRESHOLD EXECEEDED
                        ;*         3            DUMP ENTIRE SECTOR BEFORE BEGINNING RETRY SEQUENCE
                        ;*         1            INHIBIT SOFTWARE DATA COMPARISONS
```

```
                        .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
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
```

```
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .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
000240                  APRIOR= PR5                     ;RK611 PRIOIRITY
120210                  AVECT1= 120210                  ;RK611 VECTOR
```

```
1174            177440          ABASE=  177440                  ;RK611 BASE
1175            000114          MEMVEC= 114                     ;VECTOR FOR MEMORY CHECK ENABLE
1176            172100          MEMBAS= 172100                  ;BUS ADDRESS FOR MEMORY CHECK ENABLE
1177            000001          PAR.EN= 1                       ;MEMORY ENABLE MEMORY CHECKING
1178
1179                            .SBTTL  RK06-RK07 CONTROLLER REGISTER DEFINITION
1180
1181            000000          RKCS1=  0                       ;CONTROL AND STATUS REGISTER 1
1182            000002          RKWC=   2                       ;WORD COUNT REGISTER
1183            000004          RKBA=   4                       ;BUS ADDRESS REGISTER
1184            000006          RKDA=   6                       ;DESIRED TRACK SECTOR REGISTER
1185            000010          RKCS2=  10                      ;CONTROL AND STATUS REGISTER 2
1186            000012          RKDS=   12                      ;DRIVE STATUS REGISTER
1187            000014          RKER=   14                      ;ERROR REGISTER
1188            000016          RKASOF= 16                      ;ATTENTION SUMMARY AND OFFSET REGISTER
1189            000020          RKDC=   20                      ;DESIRED CYLINDER REGISTER
1190            000020          RKDCYL= 20                      ;DESIRED CYLINDER REGISTER
1191            000024          RKDB=   24                      ;DATA BUFFER
1192            000026          RKMR1=  26                      ;MAINTENANCE REGISTER 1
1193            000034          RKMR2=  34                      ;MAINTENANCE REGISTER 2
1194            000036          RKMR3=  36                      ;MAINTENANCE REGISTER 3
1195            000030          RKPOS=  30                      ;ECC POSITION INFORMATION
1196            000030          RKECPS= 30                      ;ECC POSITION INFORMATION
1197            000032          RKPAT=  32                      ;ECC PATTERN INFORMATION
1198            000032          RKECPT= 32                      ;ECC PATTERN INFORMATION
1199
1200                            .SBTTL  DRIVE COMMANDS
1201
1202            000101          SELDRV= 101                     ;SELECT DRIVE
1203            000103          PACK=   103                     ;PACK ACKNOWLEDGE
1204            000105          CLEAR=  105                     ;DRIVE CLEAR
1205            000107          UNLOAD= 107                     ;UNLOAD
1206            000111          SRTSPL= 111                     ;START SPINDLE
1207            000113          RECAL=  113                     ;RECALIBRATE
1208            000115          OFFSET= 115                     ;OFFSET
1209            000117          SEEK=   117                     ;SEEK
1210            000121          RDDATA= 121                     ;READ DATA
1211            000123          WRDATA= 123                     ;WRITE DATA
1212            000125          RDHEAD= 125                     ;READ HEADER
1213            000127          WRHEAD= 127                     ;WRITE HEADER AND DATA
1214            000131          WRTCHK= 131                     ;WRITE CHECK
1215
1216                            ;       THE FOLLOWING ARE NOT DRIVE COMMANDS BUT ARE USED BY THE DRIVER
1217                            ;       TO SIMULATE A SPECIFIC DESIRED OPERATION
1218
1219            000140          RELEAS= 140                     ;RELEASE DRIVE
1220            000141          RDSTAT= 141                     ;GET ALL STATUS FROM DRIVE
1221            000164          RDALHD= 164                     ;READ ALL HEADERS
1222            000176          CONCLR= 176                     ;CONTROLLER CLEAR (BIT 15 OF CS1)
1223            000177          SUBCLR= 177                     ;SUBSYSTEM CLEAR (BIT 5 OF CS2)
1224            000300          INTR=   300                     ;GENERATE INTERRUPT TO CPU
1225
1226                            ;       DRIVER ISSUED SERVICE COMMANDS
1227
1228            000001          DR.SEL= 001                     ;DRIVE SELECT
1229            000005          DR.CLR= 005                     ;DRIVE CLEAR
```

```
1230
1231                                    .SBTTL  CONTROL AND STATUS REGISTER 1 BITS
1232
1233    000001                          GO=     BIT0                    ;GO BIT
1234    000100                          IE=     BIT6                    ;INTERRUPT ENABLE
1235    000200                          RDY=    BIT7                    ;CONTROLLER READY
1236    000400                          BA16=   BIT8                    ;BUS ADDRESS BIT 16
1237    001000                          BA17=   BIT9                    ;BUS ADDRESS BIT 17
1238    002000                          CDT=    BIT10                   ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1239    004000                          CTO=    BIT11                   ;CONTROLLER TIMED OUT WAITING FOR
1240                                                                    ;  DRIVE RESPONSE
1241    010000                          CFMT=   BIT12                   ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1242    020000                          SPAR=   BIT13                   ;DRIVE BUS PARITY ERROR DETECTED BY CONTROLLER
1243    040000                          DI=     BIT14                   ;DRIVE INTERRUPT
1244    100000                          CERR=   BIT15                   ;CONTROLLER ERROR
1245    100000                          CCLR=   BIT15                   ;CONTROLLER CLEAR
1246
1247                                    ;       THESE BIT DEFINITIONS ARE USED FOR ADDRESS
1248                                    ;       THE HIGH BYTE OF RKCS1
1249
1250    000001                          B.BA16= BIT0                    ;BUS ADDRESS BIT 16
1251    000002                          B.BA17= BIT1                    ;BUS ADDRESS BIT 17
1252    000004                          B.CDT=  BIT2                    ;CONTROLLER DRIVE TYPE (0=RK06, 1=RK07)
1253    000020                          B.CFMT= BIT4                    ;CONTROLLER DRIVE FORMAT (0=22 SECTOR, 1=20 SECTOR)
1254
1255                                    .SBTTL  CONTROL AND STATUS REGISTER 2 BITS
1256
1257    000007                          DRVMSK= 7                       ;MASK FOR DRIVE SELECTION CODE
1258    000010                          DESL=   BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1259    000010                          RLS=    BIT3                    ;DESELECT OR RELEASE DRIVE IN BITS 0-2
1260    000020                          BAI=    BIT4                    ;BUS ADDRESS INCREMENT INHIBIT
1261    000040                          CLR=    BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1262    000040                          SCLR=   BIT5                    ;CLEAR CONTROLLER AND ALL DRIVES
1263    000100                          IR=     BIT6                    ;INPUT READY
1264    000200                          OR=     BIT7                    ;OUTPUT READY
1265    000400                          UFE=    BIT8                    ;UNIT FIELD ERROR
1266    001000                          MDS=    BIT9                    ;MULTIPLE DRIVE SELECT
1267    002000                          PGE=    BIT10                   ;PROGRAMMING ERROR
1268    004000                          NEM=    BIT11                   ;NON-EXISTENT MEMORY
1269    010000                          NED=    BIT12                   ;NON-EXISTENT DRIVE
1270    020000                          UPE=    BIT13                   ;UNIBUS PARITY ERROR
1271    040000                          WCE=    BIT14                   ;WRITE CHECK ERROR
1272    100000                          DLT=    BIT15                   ;DATA LATE ERROR
1273
1274                                    .SBTTL  ERROR REGISTER BIT DEFINITION
1275
1276    000001                          ILC=    BIT0                    ;ILLEGAL FUNCTION CODE
1277                                    ;*ILF=  BIT0                    ;ILLEGAL FUNCTION CODE
1278    000002                          SKI=    BIT1                    ;SEEK INCOMPLETE
1279    000004                          ILF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
1280    000004                          NXF=    BIT2                    ;ILLEGAL DRIVE FUNCTION
1281    000010                          DRPAR=  BIT3                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1282    000020                          FMTE=   BIT4                    ;FORMAT ERROR
1283    000040                          DTYE=   BIT5                    ;DRIVE TYPE ERROR
1284    000100                          ECH=    BIT6                    ;ECC HARD
1285    000200                          BSE=    BIT7                    ;BAD SECTOR ERROR
```

```
                        HCRC=    BIT8                ;HEADER CRC ERRROR
                        HVRC=    BIT8                ;HEADER VRC ERRROR
                        COE=     BIT9                ;CYLINDER ADDRESS OVERFLOW ERROR
                        IDAE=    BIT10               ;INVALID DISK ADDRESS ERROR
                        WLE=     BIT11               ;WRITE LOCK ERROR
                        DTE=     BIT12               ;DRIVE TIMING ERROR
                        OPI=     BIT13               ;OPERATION (SEARCH) INCOMPLETE
                        UNS=     BIT14               ;DRIVE UNSAFE
                        DCK=     BIT15               ;DATA CHECK

                        .SBTTL  STATUS REGISTER BIT DEFINITION

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

                        .SBTTL  MAINTENANCE REGISTER 1 BIT DEFINITION

                        MESMSK=  17                  ;MESSAGE MASK

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

                        .SBTTL  DEFINITION OF DRIVE STATUS BYTE 00 MESSAGE A

                        S.DRA=   BIT5                ;DRIVE AVAILIABLE
                        S.VV=    BIT6                ;VOLUME VALID
                        S.DRY=   BIT7                ;DRIVE READY
                        S.TYPE=  BIT8                ;DRIVE TYPE
                        S.FORM=  BIT9                ;DRIVE FORMAT
                        S.OFF=   BIT10               ;OFFSET
                        S.WRL=   BIT11               ;WRITE LOCK
                        S.SPIN=  BIT12               ;SPINDLE ON
                        S.PIP=   BIT13               ;POSITIONING IN PROGRESS
```

```
1342            040000          S.DSC=  BIT14                   ;DRIVE STATUS CHANGE
1343
1344                            .SBTTL  DEFINITION OF DPIVE STATUS BYTE 00 MESSAGE B
1345
1346            000040          S.ICYL= BIT5                    ;ILLEGAL CYLINDER ADDRESS
1347            000100          S.ACLO= BIT6                    ;AC LOW
1348            000200          S.FLT=  BIT7                    ;DRIVE FAULT
1349            000400          S.ILF=  BIT8                    ;ILLEGAL FUNCTION
1350            001000          S.PAR=  BIT9                    ;DRIVE DETECTED DRIVE BUS PARITY ERROR
1351            002000          S.SKI=  BIT10                   ;SEEK INCOMPLETE
1352            004000          S.WLE=  BIT11                   ;WRITE LOCK ERROR
1353            010000          S.SPLS= BIT12                   ;SPEED LOSS
1354            010000          S.DCLO= BIT12                   ;DC LOW
1355            020000          S.DROT= BIT13                   ;DRIVE OFF TRACK
1356            040000          S.UNS=  BIT14                   ;DRIVE UNSAFE
1357
1358                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE A
1359
1360            000020          S.XDOK= BIT4                    ;TRANSDUCER OK
1361            000040          S.HDHM= BIT5                    ;HEADS HOME
1362            000100          S.BRHM= BIT6                    ;BRUSHES HOME
1363            000200          S.DOOR= BIT7                    ;DOOR INTERLOCKED
1364            000400          S.CART= BIT8                    ;CARTRAGE INTERLOCK
1365            001000          S.SPOK= BIT9                    ;SPEED OK
1366            002000          S.FWD=  BIT10                   ;FORWARD
1367            004000          S.REV=  BIT11                   ;REVERSE
1368            010000          S.LOAD= BIT12                   ;HEADS LOADING
1369            020000          S.RTZ=  BIT13                   ;RETURN TO ZERO
1370            040000          S.UNLD= BIT14                   ;HEADS UNLOADING
1371
1372                            .SBTTL  DEFINITION OF DRIVE STATUS BYTE 01 MESSAGE B
1373
1374            000020          S.SECT= BIT4                    ;SECTOR ERROR
1375            000040          S.WCLK= BIT5                    ;WRITE CLOCK AND NO WRITE GATE
1376            000100          S.WGAT= BIT6                    ;WRITE GATE AND NO TRANSISTIONS
1377            000200          S.HDFL= BIT7                    ;HEAD FAULT
1378            000400          S.MHD=  BIT8                    ;MULTIPLE HEAD SELECT
1379            001000          S.XERR= BIT9                    ;INDEX ERROR
1380            002000          S.DIB=  BIT10                   ;DIBIT ERROR
1381            004000          S.PLO=  BIT11                   ;PLO ERROR
1382            010000          S.NMOV= BIT12                   ;SEEK AND NO MOTION
1383            020000          S.LIMD= BIT13                   ;LIMIT DETECT ON SEEK
1384            040000          S.BRKE= BIT14                   ;SERVO-BRAKE
1385
1386                            .SBTTL  COMMON MASKS
1387
1388            000007          M.DRV=  7                       ;DRIVE CODE
1389            100000          M.PAR=  BIT15                   ;PARITY
1390            000003          M.ID=   3                       ;BYTE ID
1391            017760          M.CDIF= 17760                   ;CYLINDER DIFFERENCE/OFFSET
1392            017760          M.CADD= 17760                   ;CYLINDER ADDRESS
1393            077770          M.SER=  77770                   ;DRIVE SERIAL NUMBER
1394            000760          M.SECT= 760                     ;SECTOR COUNT
1395            007000          M.HEAD= 7000                    ;HEAD DECODE
1396
1397                            .SBTTL  DEFAULT DRIVE VALUES
```

```
1398
1399    000000          D.MNCL= 0                       ;MINIMUM CYLINDER
1400    000000          D.MNTR= 0                       ;MINIMUM TRACK
1401    000002          D.MXTR= 2                       ;MAXIMUM TRACK
1402    000000          D.MNSC= 0                       ;MINIMUM SECTOR
1403    000025          D.MXSC= 21.                     ;MAXIMUM SECTOR
1404    000003          D.RATE= 3                       ;READ/WRITE RATIO (5/3)
1405    000000          D.AWCK= 0                       ;NO AUTOMATIC WRITE-CHECK AFTER EVERY WRITE
1406    000012          D.CERT= 10.                     ;CORRECTABLE READ ERROR THRESHOLD
1407    000005          D.UERT= 5                       ;UNCORRECTABLE READ ERROR THRESHOLD
1408    000005          D.SKET= 5                       ;SEEK ERROR THRESHOLD
1409    077777          D.CTRH= 77777                   ;COMMAND COUNT THRESHOLD
1410    177770          D.CTRL= 177770                  ;  (MAXIMUM OF COMMANDS ISSUED TO DRIVE)
1411    077777          D.WTHI= 77777                   ;WORDS TRANSFERRED THRESHOLD
1412    177770          D.WTLO= 177770                  ;  (MAXIMUM NUMBER OF WORDS READ OR WRITTEN
1413                                                    ;  ON PACK)
1414    000020          T.NER=  20                      ;THRESHOLD OF OTHER DRIVE ERRORS
1415
1416                    .SBTTL  OFFSET VALUES
1417
1418    000060          OFFP12= 060                     ;OFFSET +1200 MICRO-INCHES
1419    000040          OFFP8=  040                     ;OFFSET +800 MICRO-INCHES
1420    000020          OFFP4=  020                     ;OFFSET +400 MICRO-INCHES
1421    000220          OFFM4=  220                     ;OFFSET -400 MICRO-INCHES
1422    000240          OFFM8=  240                     ;OFFSET -800 MICRO-INCHES
1423    000260          OFFM12= 260                     ;OFFSET -1200 MICRO-INCHES
1424
1425                    .SBTTL  DEFAULT CONTROLLER THRESHOLD
1426
1427    001000          T.DLT=  1000                    ;THRESHOLD FOR DATA LATE
1428    000012          T.CONT= 10.                     ;THRESHOLD FOR OTHER CONTROLLER ERRORS
```

```
                        .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 RK06/07 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
```

```
                                CMDTO=  BIT6                    ;NO TERMINATION TO COMMAND FOR AT
                                                                ;  LEAST 1 SECOND
                                W.WCK=  BIT7                    ;WRITE FOR WRITE WRITE CHECK
                                NOCHK=  BIT8                    ;NO CHECK, DO NOT SET INTERRUPT ENABLE
                                PBSVAL= BIT9                    ;PARAMETER STATUS WORDS VALID
                                                                ;  (SET WHEN ERROR TERMINATION OR
                                                                ;  READ STATUS COMMAND)
                                DRPDRV= BIT10                   ;DROP DRIVE FROM TEST SEQUENCE
                                NODSC=  BIT11                   ;ATTENTION SET BUT DCS AND FAULT RESET
                                DRVSZD= BIT12                   ;DRIVE SEIZED BY OTHER PORT
                                E.UNLD= BIT13                   ;DRIVE UNLOADED DUE TO ERROR
                                Q.INIT= BIT14                   ;PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
                                DTBAII= BIT15                   ;INHIBIT BUS ADDRESS INCREMENT

                                .SBTTL  PARAMETERS PASSED FROM DRIVER TO PROGRAM

                                ;       THE FOLLOWING DEFINITIONS ARE USED FOR REGISTER RETURNS
                                ;         FROM THE DRIVER TO THE CALLING PROGRAM

000016                          P.CS1=  16                      ;COMMAND AND STATUS REGISTER 1
000020                          P.CS2=  20                      ;COMMAND AND STATUS REGISTER 2
000022                          P.WCR=  22                      ;WORD COUNT REGISTER
000024                          P.BAR=  24                      ;BUS ADDRESS REGISTER
000026                          P.DTS=  26                      ;DESIRED TRACK SECTOR REGISTER
000030                          P.DCYL= 30                      ;DESIRED CYLINDER REGISTER
000032                          P.ASOF= 32                      ;ATTENTION SUMMARY/OFFSET REGISTER
000034                          P.ER=   34                      ;ERROR REGISTER
000036                          P.DS=   36                      ;STATUS REGISTER
000040                          P.A00=  40                      ;MESSAGE A STATUS BYTE 00
000042                          P.B00=  42                      ;MESSAGE B STATUS BYTE 00
000044                          P.A01=  44                      ;MESSAGE A STATUS BYTE 01
000046                          P.B01=  46                      ;MESSAGE B STATUS BYTE 01
000050                          P.A10=  50                      ;MESSGGE A STATUS BYTE 10
000052                          P.B10=  52                      ;MESSAGE B STATUS BYTE 10
000054                          P.A11=  54                      ;MESSAGE A STATUS BYTE 11
000056                          P.B11=  56                      ;MESSAGE B STATUS BYTE 11
000060                          P.EPOS= 60                      ;ECC POSITION INFORMATION
000062                          P.EPAT= 62                      ;ECC PATTERN INFORMATION

                                .SBTTL  DRIVE SPECIFIC PARAMETERS

000064                          P.QLNK= 64                      ;QUEUE LINK
```

```
                        ;       THE FOLLOWING DEFINITIONS ARE USED FOR DRIVE PARAMETER ACCESS

000066                  P.MNCL= 66                      ;MINIMUM CYLINDER
000070                  P.MXCL= 70                      ;MAXIMUM CYLINDER
000072                  P.MNTR= 72                      ;MINIMUM TRACK
000073                  P.MXTR= 73                      ;MAXIMUM TRACK
000074                  P.MNSC= 74                      ;MINIMUM SECTOR
000075                  P.MXSC= 75                      ;MAXIMUM SECTOR
000076                  P.RATE= 76                      ;READ/WRITE RATIO
000077                  P.AWCK= 77                      ;AUTOMATIC WRITE CHECK
000100                  P.CERT= 100                     ;CORRECTABLE READ ERROR THRESHOLD
000102                  P.UERT= 102                     ;UNCORRECTABLE READ ERROR THRESHOLD
000104                  P.SKET= 104                     ;SEEK ERROR THRESHOLD
000106                  P.MXCD= 106                     ;MAXIMUM COMMANDS ISSUED TO DRIVE
000112                  P.MXWT= 112                     ;MAXIMUM NUMBER WORDS TRANSFERRED
000116                  P.SMPL= 116                     ;SAMPLE COMPARE FLAG
000117                  P.ECMP= 117                     ;ECC COMPARE FLAG
000120                  P.IERC= 120                     ;INHIBIT ERROR CORRECTION
000121                  P.DPAT= 121                     ;DATA PATTERN USED

                        ;       THE FOLLOWING DEFINITIONS ARE USED IN STORING
                        ;         CURRENT RANDOMLY GENERATED COMMAND

000122                  P.RDPT= 122                     ;CURRENT RANDOMLY GENERATED DATA PATTERN
000123                  P.RCMD= 123                     ;CURRENT RANDOMLY GENERATED COMMAND
000124                  P.RCYL= 124                     ;CURRENT RANDOMLY GENERATED CYLINDER
000126                  P.RSEC= 126                     ;CURRENT RANDOMLY GENERATED SECTOR
000127                  P.RTRK= 127                     ;CURRENT RANDOMLY GENERATED TRACK
000130                  P.RWC=  130                     ;CURRENT RANDOMLY GENERATED WORD COUNT
000132                  P.RBAL= 132                     ;CURRENT RANDOMLY GENERATER BUS ADDRESS
000134                  P.RBAH= 134

                        ;       THE FOLLOWING DEFINITIONS ARE USED IN STORING
                        ;         LAST ISSUED

000135                  P.LCMD= 135                     ;LAST COMMAND
000136                  P.LCYL= 136                     ;LAST CYLINDER
000140                  P.LSEC= 140                     ;LAST SECTOR
000141                  P.LTRK= 141                     ;LAST TRACK
000142                  P.LWC=  142                     ;LAST WORD COUNT
000144                  P.LDPT= 144                     ;LAST DATA PATTERN
000145                  P.BUFF= 145                     ;BUFFER ALLOCATED
```

```
1569                                    ;       THE FOLLOWING IS USED FOR THE RETRY SEQUENCE
1570
1571            000146                  P.RECT= 146             ;RETRY COUNT
1572            000147                  P.RERD= 147             ;RETRY INDICATION FOR DATA ERRORS
1573                                                            ;0      NOT IN RETRY SEQUENCE
1574                                                            ;1-20   REREAD AT CENTER LINE
1575                                                            ;21     OFFSET +400 MICRO-INCHES
1576                                                            ;22-23  REREAD OFFSET +400 MICRO-INCHES
1577                                                            ;24     OFFSET -400 MICRO-INCHES
1578                                                            ;25-26  REREAD OFFSET -400 MICRO-INCHES
1579                                                            ;27     OFFSET +800 MICRO-INCHES
1580                                                            ;30-31  REREAD OFFSET +800 MICRO-INCHES
1581                                                            ;32     OFFSET -800 MICRO-INCHES
1582                                                            ;33-34  REREAD OFFSET -800 MICRO-INCHES
1583                                                            ;35     OFFSET +1200 MICRO-INCHES
1584                                                            ;36-37  REREAD OFFSET +1200 MICRO-INCHES
1585                                                            ;40     OFFSET -1200 MICRO-INCHES
1586                                                            ;41-42  REREAD OFFSET -1200 MICRO-INCHES
1587
1588            000150                  P.DSTT= 150             ;DRIVE STATUS INFORMATION FOR ERROR PROCESSING
1589                                                            ;BIT 0  INITIAL DATA ERROR
1590                                                            ;BIT 1  DATA ERROR BEING PROCESSED
1591                                                            ;BIT 2  NON-DATA ERROR BEING PROCESSED
1592                                                            ;BIT 3  SECTOR IN ERROR PRINTED
1593                                                            ;BIT 4  READ HEADER ISSUED
1594                                                            ;BIT 6  WAITING FOR READ STATUS FOR
1595                                                            ;       POSITIONING TIME OUT
1596                                                            ;BIT 7  FIRST ERROR
1597                                                            ;BIT 11 RECALIBRATE HAS BEEN ISSUED
1598                                                            ;BIT 12 SEEK HAS BEEN ISSUED
1599                                                            ;BIT 13 OFFSET HAS BEEN ISSUED
1600                                                            ;BIT 14 BAD SECTOR PROCESSING FOR WRITE CHECK
1601                                                            ;BIT 15 ERROR ENQUEUED
1602
1603                                    ;       THE FOLLOWING DEFINITIONS ARE USED IN STATISTICAL GATHERING
1604
1605            000152                  P.NODR= 152             ;NUMBER OF ORDERS (2 WORDS)
1606            000156                  P.NWRT= 156             ;NUMBER OR WORDS WRITTEN (3 WORDS)
1607            000164                  P.NRD=  164             ;NUMBER OF WORDS READ (3 WORDS)
1608            000172                  P.SRRD= 172             ;NUMBER SOFT ERRORS (REREAD CORRECTABLE)
1609            000174                  P.SOFF= 174             ;NUMBER SOFT ERRORS (OFFSET CORRECTABLE)
1610            000176                  P.SECC= 176             ;NUMBER SOFT ERRORS (ECC CORRECTABLE)
1611            000200                  P.HARD= 200             ;NUMBER OF HARD DATA ERRORS
1612            000202                  P.NSKI= 202             ;NUMBER OF SEEK INCOMPLETE
1613            000204                  P.NOPI= 204             ;NUMBER OF OPERATION INCOMPLETES
1614            000206                  P.NER=  206             ;NUMBER OF ALL OTHER ERRORS
```

```
1615                                    ;       THE FOLLOWING IS USED FOR THE DRIVE ASSIGNMENT SEQUENCE
1616
1617    000210                          P.ASSN= 210                     ;ASSIGNMENT COMMAND SEQUENCE
1618                                                                    ;0      INITIAL STATE
1619                                                                    ;BIT 0  DRIVE SERIAL NUMBER
1620                                                                    ;BIT 1  START SPINDLE
1621                                                                    ;BIT 2  PACK ACKNOWLEDGE
1622                                                                    ;BIT 3  READ PACK SERIAL NUMBER
1623                                                                    ;BIT 4  WRITE PACK COMMAND ISSUED
1624                                                                    ;BIT 5  END OF PACK WRITE
1625                                                                    ;BIT 6  RECAL
1626                                                                    ;BIT 7  RECAL AND RETRY READ PACK
1627                                                                    ;       SERIAL NUMBER
1628
1629    000211                          P.SEEK= 211                     ;SEEK TO PREVIOUS CYLINDER FLAG
1630
1631                                    ;       THE FOLLOWING LOCATION IS USED TO INDICATE ERRORS
1632
1633    000212                          P.ERR=  212                     ;BIT 2  DATA COMPARISON ERROR
1634                                                                    ;BIT 3  DRIVE ERROR NOT REPORTED
1635                                                                    ;BIT 4  DRIVE SEIZED TIME OUT
1636                                                                    ;BIT 5  DRIVE POSITIONING TIME OUT
1637                                                                    ;BIT 6  ERROR WHILE ENQUEUED
1638                                                                    ;BIT 7  ATTENTION WHEN DRIVE NOT IN USE
1639                                                                    ;BIT 8  TIME OUT WHILE WAITING FOR HEADS
1640                                                                    ;       TO LOAD AFTER ERROR
1641                                                                    ;BIT 10 DRIVE NOT READY AFTER START SPINDLE
1642                                                                    ;BIT 11 PACK ACKNOWLEDGE DID NOT SET VOLUME
1643                                                                    ;       VALID
1644
1645    000214                          P.ERHD= 214                     ;HEADER ADDRESS OF FIRST DATA MISCOMPARE (3 BYTES)
1646    000217                          P.ERAD= 217                     ;OFFSET COUNT FROM HEADER ADDRESS
1647    000220                          P.CMLG= 220                     ;COMPARISON LENGTH
1648    000222                          P.BFCP= 222                     ;BUFFER COMPARE LENGTH
1649
1650                                    ;       THE FOLLOWING LOCATIONS ARE USED FOR DRIVE SERIAL NUMBER,
1651                                    ;         PACK SERIAL NUMBER, AND TRACK EXCLUSIONS
1652
1653    000224                          P.SERL= 224                     ;DRIVE SERIAL NUMBER
1654    000226                          P.PKSR= 226                     ;CARTRIAGE SERIAL NUMBER
1655    000232                          P.EXAR= 232                     ;EXCLUDED AREA INFORMATION
```

```
                                .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)
000200  000137  006670          JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
000204  000137  006652          JMP     RESTRT          ;BRANCH TO RESTART
                        000214          .=214
000214  000137  006642          JMP     PARSRT          ;GET SYSTEM PARAMETERS AND START TESTING
000220  000137  065062          JMP     ..DUMP          ;  *** MEMORY DUMP (DEBUG ONLY)
000224  000700        ..LOW:    700                     ;START OF DUMP
000226  004600        ..HIGH:   PATTAB-2                ;LAST ADDRESS OF DUMP
                        000240          .=240
000240  000137  006662          JMP     START1
                        001000          .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;*************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*************************************************************
                        001000          .$X=.   ;;SAVE CURRENT LOCATION
                        000024          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200          200     ;;FOR APT START UP
                        000044          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000          $APTHDR ;;POINT TO APT HEADER BLOCK
                        001000          .=.$X   ;;RESET LOCATION COUNTER
                                ;;*************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                  $APTHD:
001000  000000          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001170          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  003000          $TSTM:  .WORD   3000    ;;RUN TIM OF LONGEST TEST
001006  003000          $PASTM: .WORD   3000    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  003000          $UNITM: .WORD   3000    ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000052                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .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  177607  000377          $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001164  077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
001165  015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001166  000012                  $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001170                          $MAIL:                          ;;APT MAILBOX
001170  000000                  $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001172  000000                  $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001174  000000                  $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001176  000000                  $PASS:  .WORD   APASS           ;;PASS COUNT
001200  000000                  $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001202  000000                  $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001204  000000                  $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001206  000000                  $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001210                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
```

```
1754  001210  000          $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1755  001211  000          $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1756  001212  000000       $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1757  001214  000000       $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1758  001216  000000       $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1759                                               BITS 15-11=CPU TYPE
1760                                               ;*
1761                                               ;*
1762                                               ;*
1763                                               ;*
1764                                               ;*
1765  001220  000          $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1766  001221  000          $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1767                                               MEM.TYPE BYTE
1768                                               ;*
1769                                               ;*
1770                                               ;*
1771  001222  000000       $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1772                                               ;*
1773  001224  000          $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1774  001225  000          $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1775  001226  000000       $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1776  001230  000          $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1777  001231  000          $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1778  001232  000000       $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1779  001234  000          $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1780  001235  000          $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1781  001236  000000       $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1782  001240  120210       $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1783  001242  000000       $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1784  001244  177440       $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1785  001246  000000       $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1786  001250  000000       $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
1787  001252  000000       $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
1788  001254  000000       $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
1789  001256  000000       $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
1790  001260  000000       $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
1791  001262  000000       $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
1792  001264  000000       $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
1793  001266  000000       $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
1794  001270  000000       $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
1795  001272  000000       $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
1796  001274  000000       $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
1797  001276  000000       $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
1798  001300  000000       $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
1799  001302  000000       $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
1800  001304  000000       $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
1801  001306  000000       $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
1802  001310  000000       $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
1803  001312  000000       $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
1804
1805
1806  001314               $ETEND:
1807
```

BITS 15-11=CPU TYPE
11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
11/70=06,PDQ=07,Q=10
BIT 10=REAL TIME CLOCK
BIT 9=FLOATING POINT PROCESSOR
BIT 8=MEMORY MANAGEMENT

MEM.TYPE BYTE -- (HIGH BYTE)
900 NSEC CORE=001
300 NSEC BIPOLAR=002
500 NSEC MOS=003

MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE

```
                                .SBTTL  QUEUE HEAD AND TAIL ALLOCATIONS

001314  000000  000000         AVAILQ: .WORD   0,0              ;HEAD AND TAIL OF AVAILABLE QUEUE
001320  000000  000000         BWAITQ: .WORD   0,0              ;HEAD AND TAIL OF BUFFER WAIT QUEUE
001324  000000  000000         CINITQ: .WORD   0,0              ;HEAD AND TAIL OF COMMAND INITIATION QUEUE
001330  000000  000000         ERRPRQ: .WORD   0,0              ;HEAD AND TAIL OF ERROR PROCESSING QUEUE

                                .SBTTL  SYSTEM PARAMETERS

001334  172540                 $LKCSR: .WORD   172540           ;ADDRESS OF KW11-P STATUS REGISTTER
001336  172542                 $LKCSB: .WORD   172542           ;ADDRESS OF KW11-P COUNTER BUFFER
001340  000104                 $LPVEC: .WORD   104              ;KW11-P VECTOR ADDRESS
001342  177546                 $LKS:   .WORD   177546           ;ADDRESS OF KW11-L STATUS REGISTER
001344  000100                 $LLVEC: .WORD   100              ;KW11-L VECTOR ADDRESS
001346  177514                 $LSCS:  .WORD   177514           ;ADDRESS OF PRINTER STATUS WORD
001350  177516                 $LSDB:  .WORD   177516           ;ADDRESS OF PRINTER DATA BUFFER
001352     074                 HZ:     .BYTE   60.              ;74 (60 DECIMAL) IF SYSTEM IS 60 HZ.
                                                                ;62 (50 DECIMAL) IF SYSTEM IS 50 HZ.
001353     000                 CLKFRQ: .BYTE   0                ;COUNT FOR SECOND
001354  013400                 MAXBUF: .WORD   256.*23.         ;MAXIMUM BUFFER FOR READ OR WRITE
001356  177777                 PASCNT: .WORD   177777           ;PASS COUNT
001360     000                 PERINV: .BYTE   0                ;INTERVAL BETWEEN PERFORMANCE SUMMARIES
001361     000                 FLAG:   .BYTE   0                ;RESTART AND PARAMETER FLAG
001362  000003                 SOFCMP: .WORD   3                ;NUMBER OF SOFTWARE COMPARES
001364     000                 CPUID:  .BYTE   0                ;CPU DESIGNATION FOR PACK WRITING WITH DUAL ACCESS
001365     000                 OVLYLD: .BYTE   0                ;OVERLAY LOADER
001366  000000                 HOUR:   .WORD   0                ;TIMER HOUR
001370  000000                 MINUTE: .WORD   0                ;TIMER MINUTE
001372  000000                 SECOND: .WORD   0                ;TIMER SECOND
001374     000                 CLKFLG: .BYTE   0                ;KW11-P OR KW11-L PRESENT
001375     000                 PERIOD: .BYTE   0                ;PERIOD SINCE LAST SATISTIC PRINT OUT
001376  000000                 TIMHR:  .WORD   0                ;HOUR COUNT FOR PRINT OUT
001400  000000                 TIMMIN: .WORD   0                ;MINUTE COUNT FOR PRINT OUT
001402  000000                 TIMSEC: .WORD   0                ;SECOND COUNT FOR PRINT OUT
001404  000000                 SAVSWR: .WORD   0                ;SAVED SWITCH REGISTER FROM POWER FAIL
001406  000000                 RELFLG: .WORD   0        ;FLAG FOR RELEASE DURING WRITE PACK

                                .SBTTL  TEMPORARY CONTROLLER REGISTER STORAGE

001410  000000                 T.CS1:  .WORD   0                ;TEMPORARY STORAGE FOR COMMAND AND STATUS
                                                                ;  REGISTER 1
001412  000000                 T.CS2:  .WORD   0                ;TEMPORARY STORAGE FOR COMMAND AND STATUS
                                                                ;  REGISTER 2
001414  000000                 T.WCR:  .WORD   0                ;TEMPORARY STORAGE FOR WORD COUNT REGISTER
001416  000000                 T.BA:   .WORD   0                ;TEMPORARY STORAGE FOR BUS ADDRESS REGISTER
001420  000000                 T.DA:   .WORD   0                ;TEMPORARY STORAGE FOR DISK TRACK AND SECTOR
001422  000000                 T.DC:   .WORD   0                ;TEMPORARY STORAGE FOR DRIVE CYLINDER
001424  000000                 T.ASOF: .WORD   0                ;TEMPORARY STORAGE FOR ATTENTION SUMMARY
                                                                ;  AND OFFSET
001426  000000                 T.ER:   .WORD   0                ;TEMPORARY STORAGE FOR ERROR REGISTER
001430  000000                 T.DS:   .WORD   0                ;TEMPORARY STORAGE FOR DRIVE STATUS REGISTER
001432  000000                 T.MR1:  .WORD   0                ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 1
001434  000000                 T.MR2:  .WORD   0                ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 2
001436  000000                 T.MR3:  .WORD   0                ;TEMPORARY STORAGE FOR MAINTENANCE REGISTER 3
001440  000000                 T.POS:  .WORD   0                ;TEMPORARY STORAGE FOR ECC POSITION
001442  000000                 T.PAT:  .WORD   0                ;TEMPORARY STORAGE FOR ECC PATTERN
```

```
1864  001444  000000                  T.DB:   .WORD   0               ;TEMPORARY STORAGE FOR DATA BUFFER REGISTER
1865
1866                                  .SBTTL  DRIVER PARAMERTERS
1867
1868  001446  177440                  RKBAS:  .WORD   177440          ;ADDRESS OF RK611 UNIBUS ADDRESS BLOCK
1869  001450  000210                  RKVEC:  .WORD   210             ;ADDRESS OF R611 VECTOR
1870  001452  000240                  RKPRI:  .WORD   PR5             ;RK611 INTERRUPT PRIORITY
1871  001454  024222                  A.NORM: NORMAL                  ;ADDRESS OF NORMAL RETURN FROM DRIVER
1872  001456  025312                  A.ABNL: ABNORM                  ;ADDRESS OF ABNORMAL RETURN FROM DRIVER
1873  001460  026346                  A.CONT: CONTRL                  ;ADDRESS OF CONTROLLER ERROR RETURN
1874  001462  000000                  E.CONT: .WORD   0               ;CONTROLLER ERROR STATUS
1875                                                                  ;  THIS LOCATION IS CLEARED WHEN EVERY COMMAND
1876                                                                  ;  IS INITIATED.  IF A CONTROLLER ERROR
1877                                                                  ;  OCCURS THE FOLLOWING BIT ASSIGNMENT IS
1878                                                                  ;  USED:
1879                                                                  ;
1880          000001                  E.CCLR= BIT0                    ;CLEAR CONTROLLER DID NOT CLEAR ERROR
1881          000002                  E.NOAT= BIT1                    ;NO ATTENTION IN ATTENTION SUMMARY REG
1882          000004                  E.UATT= BIT2                    ;UNSOLICATED ATTENTION (SEQUENTIAL ONLY)
1883          000010                  E.UDAT= BIT3                    ;UNEXPECTED DATA TYPE ERROR
1884          000020                  E.CLAT= BIT4                    ;ATTENTION DID NOT RESET WITH CLEAR
1885          000040                  E.SCLR= BIT5                    ;SUBSYSTEM CLEAR DID NOT CLEAR DRIVE
1886                                                                  ;  ATTENTION
1887          000100                  E.ILLD= BIT6                    ;ILLEGAL DRIVER COMMAND
1888          000400                  E.DLT=  BIT8                    ;DATA LATE WHEN UNLOADING HEADER
1889          001000                  E.CERR= BIT9                    ;CONTROLLER ERROR DURING DRIVER SERVICING
1890          002000                  E.DPAR= BIT10                   ;DRIVE DETECTED PARITY ERROR
1891          040000                  E.CMTO= BIT14                   ;CONTROLLER COMMAND TIME OUT (QUEUED ONLY)
1892          100000                  E.MDS=  BIT15                   ;MULTIPLE DRIVE SELECT
1893
1894  001464  000000                  O.WAIT: .WORD   0               ;PARAMETER BLOCK OF THE DRIVE
1895                                                                  ;WAITING FOR COMMAND COMPLETION
1896  001466  000400                  W.MTIM: .WORD   400             ;LOOP COUNTER FOR MILISECOND SCAN OF DRIVE
1897  001470  000400                  W.MILI: .WORD   400             ;16 MILISECOND TIME FOR PROGRAM
1898                                                                  ;
1899                                                                  ;       CPU     VALUE
1900                                                                  ;       ---     -----
1901                                                                  ;
1902                                                                  ;       11/05   100
1903                                                                  ;       11/10
1904                                                                  ;       11/20
1905                                                                  ;       11/34
1906                                                                  ;       11/40
1907                                                                  ;       11/45   400
1908                                                                  ;       11/50
1909                                                                  ;       11/70
1910  001472  000300                  W.SEC:  .WORD   300             ;SECOND COUNT COUNT FOR ALL COMMANDS
1911                                                                  ;  EXCEPT START SPINDLE
1912  001474  003000                  W.8SEC: .WORD   3000            ;8 SECOND FOR DRIVE CYCLE DOWN
1913  001476  030000                  W.MIN:  .WORD   30000           ;MINUTE TIME FOR START SPINDLE
1914  001500  000000                  OPTFLG: .WORD   0               ;FLAG = 1 WHILE IN CMND OPTIMIZER
1915  001502  000000                  HDR.AD: .WORD   0               ;ADDRESS USED FOR READ ALL HEADERS
1916  001504  000000                  HDR.CT: .WORD   0               ;NUMBER OF HEADERS LEFT TO READ FOR READ
1917                                                                  ;  ALL HEADERS
1918  001506      000                 I.ISRL: .BYTE   0               ;INTERRUPT OR RELEASED COMMAND ISSUED
1919  001507      002     004     010 H.HEAD: .BYTE   2,4,10          ;HEAD DECODES
```

```
1920  001512  000     W.TIME: .BYTE   0               ;DRIVES BEING WATCH-DOG TIMED
1921  001513  377     O.OVER: .BYTE   -1              ;OVERLAPPED OPERATIONS
1922
1923                  .SBTTL  TABLE OF INTERRUPT MASKS
1924
1925  001514  001     INTMSK: .BYTE   1               ;INTERRUPT FOR DRIVE 0
1926  001515  002             .BYTE   2               ;INTERRUPT FOR DRIVE 1
1927  001516  004             .BYTE   4               ;INTERRUPT FOR DRIVE 2
1928  001517  010             .BYTE   10              ;INTERRUPT FOR DRIVE 3
1929  001520  020             .BYTE   20              ;INTERRUPT FOR DRIVE 4
1930  001521  040             .BYTE   40              ;INTERRUPT FOR DRIVE 5
1931  001522  100             .BYTE   100             ;INTERRUPT FOR DRIVE 6
1932  001523  200             .BYTE   200             ;INTERRUPT FOR DRIVE 7
1933
1934                  .SBTTL  PARAMETER BLOCK TABLE
1935
1936  001524  001736  PBLKT:  PARM0                   ;ADDRESS OF DRIVE 0'S PARAMETER BLOCK
1937  001526  002214          PARM1                   ;ADDRESS OF DRIVE 1'S PARAMETER BLOCK
1938  001530  002472          PARM2                   ;ADDRESS OF DRIVE 2'S PARAMETER BLOCK
1939  001532  002750          PARM3                   ;ADDRESS OF DRIVE 3'S PARAMETER BLOCK
1940  001534  003226          PARM4                   ;ADDRESS OF DRIVE 4'S PARAMETER BLOCK
1941  001536  003504          PARM5                   ;ADDRESS OF DRIVE 5'S PARAMETER BLOCK
1942  001540  003762          PARM6                   ;ADDRESS OF DRIVE 6'S PARAMETER BLOCK
1943  001542  004240          PARM7                   ;ADDRESS OF DRIVE 7'S PARAMETER BLOCK
1944
1945                  .SBTTL  WATCH-DOG TIMER COUNTS
1946
1947  001544  000000  W.DRV:  .WORD   0               ;WATCH-DOG FOR DRIVE 0
1948  001546  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 1
1949  001550  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 2
1950  001552  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 3
1951  001554  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 4
1952  001556  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 5
1953  001560  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 6
1954  001562  000000          .WORD   0               ;WATCH-DOG FOR DRIVE 7
```

```
                        .SBTTL  PERFORMANCE EXERCISER STATUS INFORMATION

001564  000000          ERCONT: .WORD   0               ;CONTROLLER ERROR (RECOVERABLE)
                                                        ;BIT 0  CONTROLLER ERROR NOT FLAGGED
                                                        ;BIT 1  WORD COUNT NOT EQUAL 0
                                                        ;BIT 2  BUS ADDRESS INCORRECT
                                                        ;BIT 3  CYLINDER, TRACK, SECTOR INCORRECT
                                                        ;BIT 15 CONTROLLER ERROR DETECTED

001566  000000          ERRPRO: .WORD   0               ;ADDRESS OF PARAMETER BLOCK PROCESSING ERROR
001570  000000          TMPCNT: .WORD   0               ;TRANSFER COUNT CALCULATIONS
001572  000000  000000  TMPTRK: .WORD   0,0             ;RANDOM TRACK CALCUATIONS
001576  000000  000000  TMPSEC: .WORD   0,0             ;RANDOM SECTOR CALCULATIONS
001602  000000  000000  PARCYL: .WORD   0,0             ;PARAMETER CYLINDER MODIFICATION
001606  000000  000000  PARTRK: .WORD   0,0             ;PARAMETER TRACK MODIFICATION
001612  000000  000000  PARSEC: .WORD   0,0             ;PARAMETER SECTOR MODIFICATION
001616  000000  000000  EXAREA: .WORD   0,0             ;EXCLUDED AREA CALCULATIONS
001622  000000  000000  ECCPAT: .WORD   0,0             ;ECC PATTERN FOR ECC CORRECTION
001626  000000          DLTCNT: .WORD   0               ;DATA LATE COUNT
001630  000000          CNTCNT: .WORD   0               ;OTHER CONTROLLER ERRORS COUNT
001632      000         DRVCNT: .BYTE   0               ;DRIVE COUNT FOR OPERATOR COMMAND
001633      000         STATIS: .BYTE   0               ;REPORT INTERVAL STATISTICS
001634  000000          RTYBA:  .WORD   0               ;RETRY BUS ADDRESS
001636  000000          RTYWC:  .WORD   0               ;RETRY WORD COUNT
001640      000         RTYSEC: .BYTE   0               ;SECTOR ADDRESS RETRY
001641      000         RTYTRK: .BYTE   0               ;TRACK ADDRESS RETRY
001642  000000          RTYCYL: .WORD   0               ;CYLINDER ADDRESS RETRY
001644      000         RTYSCN: .BYTE   0               ;NUMBER OF SECTORS TO BE TRANSFERRED IN RETRY
001645      000         RTYCMD: .BYTE   0               ;COMMAND BEING RETRIED
001646  000000          HEAD1:  .WORD   0               ;TEMPORARY STORAGE FOR FIRST WORD OF HEADER
001650  000000          HEAD2:  .WORD   0               ;TEMPORARY STORAGE FOR SECOND WORD OF HEADER
001652  000000          HEAD3:  .WORD   0               ;TEMPORARY STORAGE FOR THIRD WORD OF HEADER
001654      000         BSSECT: .BYTE   0               ;SECTOR IN BAD SECTOR SERVICING
001655      000         BSTRCK: .BYTE   0               ;TRACK IN BAD SECTOR SEVICING
001656  000000          BSCYLN: .WORD   0               ;CYLINDER IN BAD SECTOR SERVICING
001660  000000          BSWC:   .WORD   0               ;WORD COUNT IN BAD SECTOR SERVICING
001662  000000          BSBA:   .WORD   0               ;BUS ADDRESS IN BAD SECTOR SERVICING
001664  000000          BSWORD: .WORD   0               ;WORDS TRANSFERRED IN BAD SECTOR SERVICING
001666  000000          POSCMD: .WORD   0               ;POSITIONING TIME OUT COMMAND

                        .SBTTL  FREE SPACE BLOCK

001670  000001          FBLKC:  .WORD   1               ;FREE BLOCK COUNT

001672  000000  000000  FBLKT:  .WORD   0,0             ;FREE BLOCK 0
001676  000000  000000          .WORD   0,0             ;FREE BLOCK 1
001702  000000  000000          .WORD   0,0             ;FREE BLOCK 2
001706  000000  000000          .WORD   0,0             ;FREE BLOCK 3
001712  000000  000000          .WORD   0,0             ;FREE BLOCK 4
001716  000000  000000          .WORD   0,0             ;FREE BLOCK 5
001722  000000  000000          .WORD   0,0             ;FREE BLOCK 6
001726  000000  000000          .WORD   0,0             ;FREE BLOCK 7
001732  000000  000000          .WORD   0,0             ;FREE BLOCK 8
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 0

001736  000             PARM0:  .BYTE   0               ;DRIVE 0
001737  000                     .BYTE   0               ;COMMAND
001740  000000                  .WORD   0               ;CYLINDER ADDRESS
001742  000                     .BYTE   0               ;SECTOR ADDRESS
001743  000                     .BYTE   0               ;TRACK ADDRESS
001744  000                     .BYTE   0               ;OFFSET VALUE
001745  000                     .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
001746  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
001750  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
001752  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
001754  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
001756  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
001760  000000                  .WORD   0               ;WORD COUNT REGISTER
001762  000000                  .WORD   0               ;BUS ADDRESS REGISTER
001764  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
001766  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
001770  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
001772  000000                  .WORD   0               ;ERROR REGISTER
001774  000000                  .WORD   0               ;STATUS REGISTER
001776  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
002000  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
002002  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
002004  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
002006  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
002010  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
002012  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
002014  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
002016  000000                  .WORD   0               ;ECC POSITION INFORMATION
002020  000000                  .WORD   0               ;ECC PATTERN INFORMATION
002022  000000                  .WORD   0               ;QUEUE LINK
002024  000000                  .WORD   0               ;MINIMUM CYLINDER
002026  000632                  .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
002030  000                     .BYTE   0               ;MINIMUM TRACK
002031  002                     .BYTE   2               ;MAXIMUM TRACK
002032  000                     .BYTE   0               ;MINIMUM SECTOR
002033  025                     .BYTE   21.             ;MAXIMUM SECTOR
002034  003                     .BYTE   3               ;READ/WRITE RATIO
002035  000                     .BYTE   0               ;AUTOMATIC WRITE CHECK
002036  000012                  .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
002040  000005                  .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
002042  000005                  .WORD   5               ;SEEK ERROR THRESHOLD
002044  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
002050  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
002054  000                     .BYTE   0               ;SAMPLE COMPARE FLAG
002055  000                     .BYTE   0               ;ECC COMPARE FLAG
002056  000                     .BYTE   0               ;INHIBIT ERROR CORRECTION
002057  000                     .BYTE   0               ;DATA PATTERN USED
002060  000                     .BYTE   0               ;CURRENT GENERATED DATA PATTERN
002061  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
002062  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
002064  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
002065  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
002066  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2064  002070  000000                    .WORD   0           ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2065  002072     000                    .BYTE   0
2066  002073     000                    .BYTE   0           ;LAST COMMAND
2067  002074  000000                    .WORD   0           ;LAST CYLINDER
2068  002076     000                    .BYTE   0           ;LAST SECTOR
2069  002077     000                    .BYTE   0           ;LAST TRACK
2070  002100  000000                    .WORD   0           ;LAST WORD COUNT
2071  002102     000                    .BYTE   0           ;LAST DATA PATTERN USED
2072  002103     000                    .BYTE   0           ;BUFFER ALLOCATED
2073  002104     000                    .BYTE   0           ;RETRY COUNT
2074  002105     000                    .BYTE   0           ;REREAD STATUS
2075  002106  000000                    .WORD   0           ;DRIVE STATUS FLAGS
2076  002110  000000  000000            .WORD   0,0         ;NUMBER OF ORDERS
2077  002114  000000  000000  000000    .WORD   0,0,0       ;NUMBER OF WORDS WRITTEN
2078  002122  000000  000000  000000    .WORD   0,0,0       ;NUMBER OF WORDS READ
2079  002130  000000                    .WORD   0           ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2080  002132  000000                    .WORD   0           ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2081  002134  000000                    .WORD   0           ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2082  002136  000000                    .WORD   0           ;NUMBER OF HARD DATA ERRORS
2083  002140  000000                    .WORD   0           ;NUMBER OF SEEK INCOMPLETES
2084  002142  000000                    .WORD   0           ;NUMBER OF MISPOSITIONING ERRORS
2085  002144  000000                    .WORD   0           ;NUMBER OF ALL OTHER ERRORS
2086  002146     000                    .BYTE   0           ;ASSIGNMENT COMMAND SEQUENCE
2087  002147     000                    .BYTE   0           ;SEEK TO PREVIOUS COMMAND FLAG
2088  002150  000000                    .WORD   0           ;ERROR STATUS INFORMATION
2089  002152  000000                    .WORD   0           ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2090  002154     000                    .BYTE   0
2091  002155     000                    .BYTE   0           ;OFFSET ADDRESS FROM HEADER ADDRESS
2092  002156  000000                    .WORD   0           ;COMPARISON LENGTH
2093  002160  000000                    .WORD   0           ;BUFFER COMPARISON LENGTH
2094  002162  007777                    .WORD   007777      ;DRIVE SERIAL NUMBER
2095  002164  000000  000000            .WORD   0,0         ;CARIAGE SERIAL NUMBER
2096  002170  000000  000000            .WORD   0,0         ;FIRST EXCLUSION AREA
2097  002174  000000  000000            .WORD   0,0         ;SECOND EXCLUSION AREA
2098  002200  000000  000000            .WORD   0,0         ;THIRD EXCLUSION AREA
2099  002204  000000  000000            .WORD   0,0         ;FOURTH EXCLUSION AREA
2100  002210  000000  000000            .WORD   0,0         ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 1

002214  001             PARM1:  .BYTE   1               ;DRIVE 1
002215  000                     .BYTE   0               ;COMMAND
002216  000000                  .WORD   0               ;CYLINDER ADDRESS
002220  000                     .BYTE   0               ;SECTOR ADDRESS
002221  000                     .BYTE   0               ;TRACK ADDRESS
002222  000                     .BYTE   0               ;OFFSET VALUE
002223  000                     .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
002224  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
002226  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
002230  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
002232  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
002234  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
002236  000000                  .WORD   0               ;WORD COUNT REGISTER
002240  000000                  .WORD   0               ;BUS ADDRESS REGISTER
002242  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
002244  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
002246  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
002250  000000                  .WORD   0               ;ERROR REGISTER
002252  000000                  .WORD   0               ;STATUS REGISTER
002254  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
002256  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
002260  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
002262  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
002264  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
002266  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
002270  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
002272  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
002274  000000                  .WORD   0               ;ECC POSITION INFORMATION
002276  000000                  .WORD   0               ;ECC PATTERN INFORMATION
002300  000000                  .WORD   0               ;QUEUE LINK
002302  000000                  .WORD   0               ;MINIMUM CYLINDER
002304  000632                  .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
002306  000                     .BYTE   0               ;MINIMUM TRACK
002307  002                     .BYTE   2               ;MAXIMUM TRACK
002310  000                     .BYTE   0               ;MINIMUM SECTOR
002311  025                     .BYTE   21.             ;MAXIMUM SECTOR
002312  003                     .BYTE   3               ;READ/WRITE RATIO
002313  000                     .BYTE   0               ;AUTOMATIC WRITE CHECK
002314  000012                  .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
002316  000005                  .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
002320  000005                  .WORD   5               ;SEEK ERROR THRESHOLD
002322  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
002326  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
002332  000                     .BYTE   0               ;SAMPLE COMPARE FLAG
002333  000                     .BYTE   0               ;ECC COMPARE FLAG
002334  000                     .BYTE   0               ;INHIBIT ERROR CORRECTION
002335  000                     .BYTE   0               ;DATA PATTERN USED
002336  000                     .BYTE   0               ;CURRENT GENERATED DATA PATTERN
002337  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
002340  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
002342  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
002343  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
002344  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2157  002346  000000                        .WORD   0              ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2158  002350     000                        .BYTE   0
2159  002351     000                        .BYTE   0              ;LAST COMMAND
2160  002352  000000                        .WORD   0              ;LAST CYLINDER
2161  002354     000                        .BYTE   0              ;LAST SECTOR
2162  002355     000                        .BYTE   0              ;LAST TRACK
2163  002356  000000                        .WORD   0              ;LAST WORD COUNT
2164  002360     000                        .BYTE   0              ;LAST DATA PATTERN USED
2165  002361     000                        .BYTE   0              ;BUFFER ALLOCATED
2166  002362     000                        .BYTE   0              ;RETRY COUNT
2167  002363     000                        .BYTE   0              ;REREAD STATUS
2168  002364  000000                        .WORD   0              ;DRIVE STATUS FLAGS
2169  002366  000000  000000                .WORD   0,0            ;NUMBER OF ORDERS
2170  002372  000000  000000  000000        .WORD   0,0,0          ;NUMBER OF WORDS WRITTEN
2171  002400  000000  000000  000000        .WORD   0,0,0          ;NUMBER OF WORDS READ
2172  002406  000000                        .WORD   0              ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2173  002410  000000                        .WORD   0              ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2174  002412  000000                        .WORD   0              ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2175  002414  000000                        .WORD   0              ;NUMBER OF HARD DATA ERRORS
2176  002416  000000                        .WORD   0              ;NUMBER OF SEEK INCOMPLETES
2177  002420  000000                        .WORD   0              ;NUMBER OF MISPOSITIONING ERRORS
2178  002422  000000                        .WORD   0              ;NUMBER OF ALL OTHER ERRORS
2179  002424     000                        .BYTE   0              ;ASSIGNMENT COMMAND SEQUENCE
2180  002425     000                        .BYTE   0              ;SEEK TO PREVIOUS COMMAND FLAG
2181  002426  000000                        .WORD   0              ;ERROR STATUS INFORMATION
2182  002430  000000                        .WORD   0              ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2183  002432     000                        .BYTE   0
2184  002433     000                        .BYTE   0              ;OFFSET ADDRESS FROM HEADER ADDRESS
2185  002434  000000                        .WORD   0              ;COMPARISON LENGTH
2186  002436  000000                        .WORD   0              ;BUFFER COMPARISON LENGTH
2187  002440  007777                        .WORD   007777         ;DRIVE SERIAL NUMBER
2188  002442  000000  000000                .WORD   0,0            ;CARIAGE SERIAL NUMBER
2189  002446  000000  000000                .WORD   0,0            ;FIRST EXCLUSION AREA
2190  002452  000000  000000                .WORD   0,0            ;SECOND EXCLUSION AREA
2191  002456  000000  000000                .WORD   0,0            ;THIRD EXCLUSION AREA
2192  002462  000000  000000                .WORD   0,0            ;FOURTH EXCLUSION AREA
2193  002466  000000  000000                .WORD   0,0            ;FIFTH EXCLUSION AREA
```

```
2194     
2195     
2196     
2197   002472   002                 .SBTTL  PARAMETER BLOCK FOR DRIVE 2
2198   002473   000
2199   002474   000000   PARM2:  .BYTE   2          ;DRIVE 2
2200   002476   000             .BYTE   0          ;COMMAND
2201   002477   000             .WORD   0          ;CYLINDER ADDRESS
2202   002500   000             .BYTE   0          ;SECTOR ADDRESS
2203   002501   000             .BYTE   0          ;TRACK ADDRESS
2204   002502   000000          .BYTE   0          ;OFFSET VALUE
2205   002504   000000          .BYTE   0          ;BUS ADDRESS (BITS 16 AND 17)
2206   002506   000000          .WORD   0          ;BUS ADDRESS (BITS 0 - 15)
2207   002510   000000          .WORD   0          ;WORD COUNT (2'S COMPLEMENT)
2208   002512   000000          .WORD   0          ;PROGRAM DRIVE STATUS INFORMATION
2209   002514   000000          .WORD   0          ;COMMAND AND STATUS REGISTER 1
2210   002516   000000          .WORD   0          ;COMMAND AND STATUS REGISTER 2
2211   002520   000000          .WORD   0          ;WORD COUNT REGISTER
2212   002522   000000          .WORD   0          ;BUS ADDRESS REGISTER
2213   002524   000000          .WORD   0          ;DESIRED TRACK AND SECTOR REGISTER
2214   002526   000000          .WORD   0          ;DESIRED CYLINDER REGISTER
2215   002530   000000          .WORD   0          ;ATTENTION SUMMARY/OFFSET REGISTER
2216   002532   000000          .WORD   0          ;ERROR REGISTER
2217   002534   000000          .WORD   0          ;STATUS REGISTER
2218   002536   000000          .WORD   0          ;MESSAGE LINE A STATUS BYTE 00
2219   002540   000000          .WORD   0          ;MESSAGE LINE B STATUS BYTE 00
2220   002542   000000          .WORD   0          ;MESSAGE LINE A STATUS BYTE 01
2221   002544   000000          .WORD   0          ;MESSAGE LINE B STATUS BYTE 01
2222   002546   000000          .WORD   0          ;MESSAGE LINE A STATUS BYTE 10
2223   002550   000000          .WORD   0          ;MESSAGE LINE B STATUS BYTE 10
2224   002552   000000          .WORD   0          ;MESSAGE LINE A STATUS BYTE 11
2225   002554   000000          .WORD   0          ;MESSAGE LINE B STATUS BYTE 11
2226   002556   000000          .WORD   0          ;ECC POSITION INFORMATION
2227   002560   000000          .WORD   0          ;ECC PATTERN INFORMATION
2228   002562   000632          .WORD   410.       ;QUEUE LINK
2229   002564      000          .BYTE   0          ;MINIMUM CYLINDER
2230   002565      002          .BYTE   2          ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
2231   002566      000          .BYTE   0          ;MINIMUM TRACK
2232   002567      025          .BYTE   21.        ;MAXIMUM TRACK
2233   002570      003          .BYTE   3          ;MINIMUM SECTOR
2234   002571      000          .BYTE   0          ;MAXIMUM SECTOR
2235   002572   000012          .WORD   10.        ;READ/WRITE RATIO
2236   002574   000005          .WORD   5          ;AUTOMATIC WRITE CHECK
2237   002576   000005          .WORD   5          ;CORRECTABLE READ ERROR THRESHOLD
2238   002600   177770   177777 .WORD   177770,177777  ;UNCORRECTABLE READ ERROR THRESHOLD
2239   002604   177770   177777 .WORD   177770,177777  ;SEEK ERROR THRESHOLD
2240   002610      000          .BYTE   0          ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
2241   002611      000          .BYTE   0          ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
2242   002612      000          .BYTE   0          ;SAMPLE COMPARE FLAG
2243   002613      000          .BYTE   0          ;ECC COMPARE FLAG
2244   002614      000          .BYTE   0          ;INHIBIT ERROR CORRECTION
2245   002615      000          .BYTE   0          ;DATA PATTERN USED
2246   002616   000000          .WORD   0          ;CURRENT GENERATED DATA PATTERN
2247   002620      000          .BYTE   0          ;CURRENT RANDOMLY GENERATED COMMAND
2248   002621      000          .BYTE   0          ;CURRENT RANDOMLY GENERATED CYLINDER
2249   002622   000000          .WORD   0          ;CURRENT RANDOMLY GENERATED SECTOR
                                                ;CURRENT RANDOMLY GENERATED TRACK
                                                ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2250  002624  000000                        .WORD   0           ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2251  002626     000                        .BYTE   0
2252  002627     000                        .BYTE   0           ;LAST COMMAND
2253  002630  000000                        .WORD   0           ;LAST CYLINDER
2254  002632     000                        .BYTE   0           ;LAST SECTOR
2255  002633     000                        .BYTE   0           ;LAST TRACK
2256  002634  000000                        .WORD   0           ;LAST WORD COUNT
2257  002636     000                        .BYTE   0           ;LAST DATA PATTERN USED
2258  002637     000                        .BYTE   0           ;BUFFER ALLOCATED
2259  002640     000                        .BYTE   0           ;RETRY COUNT
2260  002641     000                        .BYTE   0           ;REREAD STATUS
2261  002642  000000                        .WORD   0           ;DRIVE STATUS FLAGS
2262  002644  000000  000000                .WORD   0,0         ;NUMBER OF ORDERS
2263  002650  000000  000000  000000        .WORD   0,0,0       ;NUMBER OF WORDS WRITTEN
2264  002656  000000  000000  000000        .WORD   0,0,0       ;NUMBER OF WORDS READ
2265  002664  000000                        .WORD   0           ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2266  002666  000000                        .WORD   0           ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2267  002670  000000                        .WORD   0           ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2268  002672  000000                        .WORD   0           ;NUMBER OF HARD DATA ERRORS
2269  002674  000000                        .WORD   0           ;NUMBER OF SEEK INCOMPLETES
2270  002676  000000                        .WORD   0           ;NUMBER OF MISPOSITIONING ERRORS
2271  002700  000000                        .WORD   0           ;NUMBER OF ALL OTHER ERRORS
2272  002702     000                        .BYTE   0           ;ASSIGNMENT COMMAND SEQUENCE
2273  002703     000                        .BYTE   0           ;SEEK TO PREVIOUS COMMAND FLAG
2274  002704  000000                        .WORD   0           ;ERROR STATUS INFORMATION
2275  002706  000000                        .WORD   0           ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2276  002710     000                        .BYTE   0
2277  002711     000                        .BYTE   0           ;OFFSET ADDRESS FROM HEADER ADDRESS
2278  002712  000000                        .WORD   0           ;COMPARISON LENGTH
2279  002714  000000                        .WORD   0           ;BUFFER COMPARISON LENGTH
2280  002716  007777                        .WORD   007777      ;DRIVE SERIAL NUMBER
2281  002720  000000  000000                .WORD   0,0         ;CARIAGE SERIAL NUMBER
2282  002724  000000  000000                .WORD   0,0         ;FIRST EXCLUSION AREA
2283  002730  000000  000000                .WORD   0,0         ;SECOND EXCLUSION AREA
2284  002734  000000  000000                .WORD   0,0         ;THIRD EXCLUSION AREA
2285  002740  000000  000000                .WORD   0,0         ;FOURTH EXCLUSION AREA
2286  002744  000000  000000                .WORD   0,0         ;FIFTH EXCLUSION AREA
```

```
                                        .SBTTL  PARAMETER BLOCK FOR DRIVE 3

 002750         003                     PARM3:  .BYTE   3                       ;DRIVE 3
 002751         000                             .BYTE   0                       ;COMMAND
 002752      000000                             .WORD   0                       ;CYLINDER ADDRESS
 002754         000                             .BYTE   0                       ;SECTOR ADDRESS
 002755         000                             .BYTE   0                       ;TRACK ADDRESS
 002756         000                             .BYTE   0                       ;OFFSET VALUE
 002757         000                             .BYTE   0                       ;BUS ADDRESS (BITS 16 AND 17)
 002760      000000                             .WORD   0                       ;BUS ADDRESS (BITS 0 - 15)
 002762      000000                             .WORD   0                       ;WORD COUNT (2'S COMPLEMENT)
 002764      000000                             .WORD   0                       ;PROGRAM DRIVE STATUS INFORMATION
 002766      000000                             .WORD   0                       ;COMMAND AND STATUS REGISTER 1
 002770      000000                             .WORD   0                       ;COMMAND AND STATUS REGISTER 2
 002772      000000                             .WORD   0                       ;WORD COUNT REGISTER
 002774      000000                             .WORD   0                       ;BUS ADDRESS REGISTER
 002776      000000                             .WORD   0                       ;DESIRED TRACK AND SECTOR REGISTER
 003000      000000                             .WORD   0                       ;DESIRED CYLINDER REGISTER
 003002      000000                             .WORD   0                       ;ATTENTION SUMMARY/OFFSET REGISTER
 003004      000000                             .WORD   0                       ;ERROR REGISTER
 003006      000000                             .WORD   0                       ;STATUS REGISTER
 003010      000000                             .WORD   0                       ;MESSAGE LINE A STATUS BYTE 00
 003012      000000                             .WORD   0                       ;MESSAGE LINE B STATUS BYTE 00
 003014      000000                             .WORD   0                       ;MESSAGE LINE A STATUS BYTE 01
 003016      000000                             .WORD   0                       ;MESSAGE LINE B STATUS BYTE 01
 003020      000000                             .WORD   0                       ;MESSAGE LINE A STATUS BYTE 10
 003022      000000                             .WORD   0                       ;MESSAGE LINE B STATUS BYTE 10
 003024      000000                             .WORD   0                       ;MESSAGE LINE A STATUS BYTE 11
 003026      000000                             .WORD   0                       ;MESSAGE LINE B STATUS BYTE 11
 003030      000000                             .WORD   0                       ;ECC POSITION INFORMATION
 003032      000000                             .WORD   0                       ;ECC PATTERN INFORMATION
 003034      000000                             .WORD   0                       ;QUEUE LINK
 003036      000000                             .WORD   0                       ;MINIMUM CYLINDER
 003040      000632                             .WORD   410.                    ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
 003042         000                             .BYTE   0                       ;MINIMUM TRACK
 003043         002                             .BYTE   2                       ;MAXIMUM TRACK
 003044         000                             .BYTE   0                       ;MINIMUM SECTOR
 003045         025                             .BYTE   21.                     ;MAXIMUM SECTOR
 003046         003                             .BYTE   3                       ;READ/WRITE RATIO
 003047         000                             .BYTE   0                       ;AUTOMATIC WRITE CHECK
 003050      000012                             .WORD   10.                     ;CORRECTABLE READ ERROR THRESHOLD
 003052      000005                             .WORD   5                       ;UNCORRECTABLE READ ERROR THRESHOLD
 003054      000005                             .WORD   5                       ;SEEK ERROR THRESHOLD
 003056      177770  177777                     .WORD   177770,177777           ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
 003062      177770  177777                     .WORD   177770,177777           ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
 003066         000                             .BYTE   0                       ;SAMPLE COMPARE FLAG
 003067         000                             .BYTE   0                       ;ECC COMPARE FLAG
 003070         000                             .BYTE   0                       ;INHIBIT ERROR CORRECTION
 003071         000                             .BYTE   0                       ;DATA PATTERN USED
 003072         000                             .BYTE   0                       ;CURRENT GENERATED DATA PATTERN
 003073         000                             .BYTE   0                       ;CURRENT RANDOMLY GENERATED COMMAND
 003074      000000                             .WORD   0                       ;CURRENT RANDOMLY GENERATED CYLINDER
 003076         000                             .BYTE   0                       ;CURRENT RANDOMLY GENERATED SECTOR
 003077         000                             .BYTE   0                       ;CURRENT RANDOMLY GENERATED TRACK
 003100      000000                             .WORD   0                       ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2343  003102  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2344  003104      000                       .BYTE   0
2345  003105      000                       .BYTE   0               ;LAST COMMAND
2346  003106  000000                        .WORD   0               ;LAST CYLINDER
2347  003110      000                       .BYTE   0               ;LAST SECTOR
2348  003111      000                       .BYTE   0               ;LAST TRACK
2349  003112  000000                        .WORD   0               ;LAST WORD COUNT
2350  003114      000                       .BYTE   0               ;LAST DATA PATTERN USED
2351  003115      000                       .BYTE   0               ;BUFFER ALLOCATED
2352  003116      000                       .BYTE   0               ;RETRY COUNT
2353  003117      000                       .BYTE   0               ;REREAD STATUS
2354  003120  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2355  003122  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2356  003126  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2357  003134  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2358  003142  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2359  003144  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2360  003146  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2361  003150  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2362  003152  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2363  003154  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2364  003156  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2365  003160      000                       .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2366  003161      000                       .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2367  003162  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2368  003164  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2369  003166      000                       .BYTE   0
2370  003167      000                       .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2371  003170  000000                        .WORD   0               ;COMPARISON LENGTH
2372  003172  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2373  003174  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2374  003176  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2375  003202  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2376  003206  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2377  003212  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2378  003216  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2379  003222  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  PARAMETER BLOCK FOR DRIVE 4

003226  004             PARM4:  .BYTE   4               ;DRIVE 4
003227  000                     .BYTE   0               ;COMMAND
003230  000000                  .WORD   0               ;CYLINDER ADDRESS
003232  000                     .BYTE   0               ;SECTOR ADDRESS
003233  000                     .BYTE   0               ;TRACK ADDRESS
003234  000                     .BYTE   0               ;OFFSET VALUE
003235  000                     .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
003236  000000                  .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
003240  000000                  .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
003242  000000                  .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
003244  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 1
003246  000000                  .WORD   0               ;COMMAND AND STATUS REGISTER 2
003250  000000                  .WORD   0               ;WORD COUNT REGISTER
003252  000000                  .WORD   0               ;BUS ADDRESS REGISTER
003254  000000                  .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
003256  000000                  .WORD   0               ;DESIRED CYLINDER REGISTER
003260  000000                  .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
003262  000000                  .WORD   0               ;ERROR REGISTER
003264  000000                  .WORD   0               ;STATUS REGISTER
003266  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
003270  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
003272  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
003274  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
003276  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
003300  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
003302  000000                  .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
003304  000000                  .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
003306  000000                  .WORD   0               ;ECC POSITION INFORMATION
003310  000000                  .WORD   0               ;ECC PATTERN INFORMATION
003312  000000                  .WORD   0               ;QUEUE LINK
003314  000000                  .WORD   0               ;MINIMUM CYLINDER
003316  000632                  .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
003320  000                     .BYTE   0               ;MINIMUM TRACK
003321  002                     .BYTE   2               ;MAXIMUM TRACK
003322  000                     .BYTE   0               ;MINIMUM SECTOR
003323  025                     .BYTE   21.             ;MAXIMUM SECTOR
003324  003                     .BYTE   3               ;READ/WRITE RATIO
003325  000                     .BYTE   0               ;AUTOMATIC WRITE CHECK
003326  000012                  .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
003330  000005                  .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
003332  000005                  .WORD   5               ;SEEK ERROR THRESHOLD
003334  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
003340  177770  177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
003344  000                     .BYTE   0               ;SAMPLE COMPARE FLAG
003345  000                     .BYTE   0               ;ECC COMPARE FLAG
003346  000                     .BYTE   0               ;INHIBIT ERROR CORRECTION
003347  000                     .BYTE   0               ;DATA PATTERN USED
003350  000                     .BYTE   0               ;CURRENT GENERATED DATA PATTERN
003351  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
003352  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
003354  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
003355  000                     .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
003356  000000                  .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2436  003360  000000                        .WORD   0          ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2437  003362     000                        .BYTE   0
2438  003363     000                        .BYTE   0          ;LAST COMMAND
2439  003364  000000                        .WORD   0          ;LAST CYLINDER
2440  003366     000                        .BYTE   0          ;LAST SECTOR
2441  003367     000                        .BYTE   0          ;LAST TRACK
2442  003370  000000                        .WORD   0          ;LAST WORD COUNT
2443  003372     000                        .BYTE   0          ;LAST DATA PATTERN USED
2444  003373     000                        .BYTE   0          ;BUFFER ALLOCATED
2445  003374     000                        .BYTE   0          ;RETRY COUNT
2446  003375     000                        .BYTE   0          ;REREAD STATUS
2447  003376  000000                        .WORD   0          ;DRIVE STATUS FLAGS
2448  003400  000000  000000                .WORD   0,0        ;NUMBER OF ORDERS
2449  003404  000000  000000  000000        .WORD   0,0,0      ;NUMBER OF WORDS WRITTEN
2450  003412  000000  000000  000000        .WORD   0,0,0      ;NUMBER OF WORDS READ
2451  003420  000000                        .WORD   0          ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2452  003422  000000                        .WORD   0          ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2453  003424  000000                        .WORD   0          ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2454  003426  000000                        .WORD   0          ;NUMBER OF HARD DATA ERRORS
2455  003430  000000                        .WORD   0          ;NUMBER OF SEEK INCOMPLETES
2456  003432  000000                        .WORD   0          ;NUMBER OF MISPOSITIONING ERRORS
2457  003434  000000                        .WORD   0          ;NUMBER OF ALL OTHER ERRORS
2458  003436     000                        .BYTE   0          ;ASSIGNMENT COMMAND SEQUENCE
2459  003437     000                        .BYTE   0          ;SEEK TO PREVIOUS COMMAND FLAG
2460  003440  000000                        .WORD   0          ;ERROR STATUS INFORMATION
2461  003442  000000                        .WORD   0          ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2462  003444     000                        .BYTE   0
2463  003445     000                        .BYTE   0          ;OFFSET ADDRESS FROM HEADER ADDRESS
2464  003446  000000                        .WORD   0          ;COMPARISON LENGTH
2465  003450  000000                        .WORD   0          ;BUFFER COMPARISON LENGTH
2466  003452  007777                        .WORD   007777     ;DRIVE SERIAL NUMBER
2467  003454  000000  000000                .WORD   0,0        ;CARIAGE SERIAL NUMBER
2468  003460  000000  000000                .WORD   0,0        ;FIRST EXCLUSION AREA
2469  003464  000000  000000                .WORD   0,0        ;SECOND EXCLUSION AREA
2470  003470  000000  000000                .WORD   0,0        ;THIRD EXCLUSION AREA
2471  003474  000000  000000                .WORD   0,0        ;FOURTH EXCLUSION AREA
2472  003500  000000  000000                .WORD   0,0        ;FIFTH EXCLUSION AREA
```

```
2473
2474                                    .SBTTL  PARAMETER BLOCK FOR DRIVE 5
2475
2476  003504      005           PARM5:  .BYTE   5               ;DRIVE 5
2477  003505      000                   .BYTE   0               ;COMMAND
2478  003506   000000                   .WORD   0               ;CYLINDER ADDRESS
2479  003510      000                   .BYTE   0               ;SECTOR ADDRESS
2480  003511      000                   .BYTE   0               ;TRACK ADDRESS
2481  003512      000                   .BYTE   0               ;OFFSET VALUE
2482  003513      000                   .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
2483  003514   000000                   .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
2484  003516   000000                   .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
2485  003520   000000                   .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
2486  003522   000000                   .WORD   0               ;COMMAND AND STATUS REGISTER 1
2487  003524   000000                   .WORD   0               ;COMMAND AND STATUS REGISTER 2
2488  003526   000000                   .WORD   0               ;WORD COUNT REGISTER
2489  003530   000000                   .WORD   0               ;BUS ADDRESS REGISTER
2490  003532   000000                   .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
2491  003534   000000                   .WORD   0               ;DESIRED CYLINDER REGISTER
2492  003536   000000                   .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
2493  003540   000000                   .WORD   0               ;ERROR REGISTER
2494  003542   000000                   .WORD   0               ;STATUS REGISTER
2495  003544   000000                   .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
2496  003546   000000                   .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
2497  003550   000000                   .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
2498  003552   000000                   .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
2499  003554   000000                   .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
2500  003556   000000                   .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
2501  003560   000000                   .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
2502  003562   000000                   .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
2503  003564   000000                   .WORD   0               ;ECC POSITION INFORMATION
2504  003566   000000                   .WORD   0               ;ECC PATTERN INFORMATION
2505  003570   000000                   .WORD   0               ;QUEUE LINK
2506  003572   000000                   .WORD   0               ;MINIMUM CYLINDER
2507  003574   000632                   .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
2508  003576      000                   .BYTE   0               ;MINIMUM TRACK
2509  003577      002                   .BYTE   2               ;MAXIMUM TRACK
2510  003600      000                   .BYTE   0               ;MINIMUM SECTOR
2511  003601      025                   .BYTE   21.             ;MAXIMUM SECTOR
2512  003602      003                   .BYTE   3               ;READ/WRITE RATIO
2513  003603      000                   .BYTE   0               ;AUTOMATIC WRITE CHECK
2514  003604   000012                   .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
2515  003606   000005                   .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
2516  003610   000005                   .WORD   5               ;SEEK ERROR THRESHOLD
2517  003612   177770   177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
2518  003616   177770   177777          .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
2519  003622      000                   .BYTE   0               ;SAMPLE COMPARE FLAG
2520  003623      000                   .BYTE   0               ;ECC COMPARE FLAG
2521  003624      000                   .BYTE   0               ;INHIBIT ERROR CORRECTION
2522  003625      000                   .BYTE   0               ;DATA PATTERN USED
2523  003626      000                   .BYTE   0               ;CURRENT GENERATED DATA PATTERN
2524  003627      000                   .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
2525  003630   000000                   .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
2526  003632      000                   .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
2527  003633      000                   .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
2528  003634   000000                   .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2529  003636  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2530  003640     000                        .BYTE   0
2531  003641     000                        .BYTE   0               ;LAST COMMAND
2532  003642  000000                        .WORD   0               ;LAST CYLINDER
2533  003644     000                        .BYTE   0               ;LAST SECTOR
2534  003645     000                        .BYTE   0               ;LAST TRACK
2535  003646  000000                        .WORD   0               ;LAST WORD COUNT
2536  003650     000                        .BYTE   0               ;LAST DATA PATTERN USED
2537  003651     000                        .BYTE   0               ;BUFFER ALLOCATED
2538  003652     000                        .BYTE   0               ;RETRY COUNT
2539  003653     000                        .BYTE   0               ;REREAD STATUS
2540  003654  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2541  003656  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2542  003662  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2543  003670  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2544  003676  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2545  003700  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2546  003702  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2547  003704  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2548  003706  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2549  003710  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2550  003712  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2551  003714     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2552  003715     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2553  003716  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2554  003720  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2555  003722     000                        .BYTE   0
2556  003723     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2557  003724  000000                        .WORD   0               ;COMPARISON LENGTH
2558  003726  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2559  003730  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2560  003732  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2561  003736  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2562  003742  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2563  003746  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2564  003752  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2565  003756  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
2566
2567                                    .SBTTL  PARAMETER BLOCK FOR DRIVE 6
2568
2569  003762  006                       PARM6:  .BYTE   6               ;DRIVE 6
2570  003763  000                               .BYTE   0               ;COMMAND
2571  003764  000000                            .WORD   0               ;CYLINDER ADDRESS
2572  003766  000                               .BYTE   0               ;SECTOR ADDRESS
2573  003767  000                               .BYTE   0               ;TRACK ADDRESS
2574  003770  000                               .BYTE   0               ;OFFSET VALUE
2575  003771  000                               .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
2576  003772  000000                            .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
2577  003774  000000                            .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
2578  003776  000000                            .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
2579  004000  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 1
2580  004002  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 2
2581  004004  000000                            .WORD   0               ;WORD COUNT REGISTER
2582  004006  000000                            .WORD   0               ;BUS ADDRESS REGISTER
2583  004010  000000                            .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
2584  004012  000000                            .WORD   0               ;DESIRED CYLINDER REGISTER
2585  004014  000000                            .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
2586  004016  000000                            .WORD   0               ;ERROR REGISTER
2587  004020  000000                            .WORD   0               ;STATUS REGISTER
2588  004022  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
2589  004024  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
2590  004026  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
2591  004030  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
2592  004032  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
2593  004034  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
2594  004036  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
2595  004040  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
2596  004042  000000                            .WORD   0               ;ECC POSITION INFORMATION
2597  004044  000000                            .WORD   0               ;ECC PATTERN INFORMATION
2598  004046  000000                            .WORD   0               ;QUEUE LINK
2599  004050  000000                            .WORD   0               ;MINIMUM CYLINDER
2600  004052  000632                            .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
2601  004054  000                               .BYTE   0               ;MINIMUM TRACK
2602  004055  002                               .BYTE   2               ;MAXIMUM TRACK
2603  004056  000                               .BYTE   0               ;MINIMUM SECTOR
2604  004057  025                               .BYTE   21.             ;MAXIMUM SECTOR
2605  004060  003                               .BYTE   3               ;READ/WRITE RATIO
2606  004061  000                               .BYTE   0               ;AUTOMATIC WRITE CHECK
2607  004062  000012                            .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
2608  004064  000005                            .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
2609  004066  000005                            .WORD   5               ;SEEK ERROR THRESHOLD
2610  004070  177770  177777                   .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
2611  004074  177770  177777                   .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
2612  004100  000                               .BYTE   0               ;SAMPLE COMPARE FLAG
2613  004101  000                               .BYTE   0               ;ECC COMPARE FLAG
2614  004102  000                               .BYTE   0               ;INHIBIT ERROR CORRECTION
2615  004103  000                               .BYTE   0               ;DATA PATTERN USED
2616  004104  000                               .BYTE   0               ;CURRENT GENERATED DATA PATTERN
2617  004105  000                               .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
2618  004106  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
2619  004110  000                               .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
2620  004111  000                               .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
2621  004112  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2622  004114  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2623  004116     000                        .BYTE   0
2624  004117     000                        .BYTE   0               ;LAST COMMAND
2625  004120  000000                        .WORD   0               ;LAST CYLINDER
2626  004122     000                        .BYTE   0               ;LAST SECTOR
2627  004123     000                        .BYTE   0               ;LAST TRACK
2628  004124  000000                        .WORD   0               ;LAST WORD COUNT
2629  004126     000                        .BYTE   0               ;LAST DATA PATTERN USED
2630  004127     000                        .BYTE   0               ;BUFFER ALLOCATED
2631  004130     000                        .BYTE   0               ;RETRY COUNT
2632  004131     000                        .BYTE   0               ;REREAD STATUS
2633  004132  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2634  004134  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2635  004140  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2636  004146  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2637  004154  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2638  004156  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2639  004160  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2640  004162  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2641  004164  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2642  004166  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2643  004170  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2644  004172     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2645  004173     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2646  004174  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2647  004176  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2648  004200     000                        .BYTE   0
2649  004201     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2650  004202  000000                        .WORD   0               ;COMPARISON LENGTH
2651  004204  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2652  004206  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2653  004210  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2654  004214  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2655  004220  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2656  004224  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2657  004230  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2658  004234  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
2659
2660                                    .SBTTL  PARAMETER BLOCK FOR DRIVE 7
2661
2662  004240    007                     PARM7:  .BYTE   7               ;DRIVE 7
2663  004241    000                             .BYTE   0               ;COMMAND
2664  004242  000000                            .WORD   0               ;CYLINDER ADDRESS
2665  004244    000                             .BYTE   0               ;SECTOR ADDRESS
2666  004245    000                             .BYTE   0               ;TRACK ADDRESS
2667  004246    000                             .BYTE   0               ;OFFSET VALUE
2668  004247    000                             .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
2669  004250  000000                            .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
2670  004252  000000                            .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
2671  004254  000000                            .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
2672  004256  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 1
2673  004260  000000                            .WORD   0               ;COMMAND AND STATUS REGISTER 2
2674  004262  000000                            .WORD   0               ;WORD COUNT REGISTER
2675  004264  000000                            .WORD   0               ;BUS ADDRESS REGISTER
2676  004266  000000                            .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
2677  004270  000000                            .WORD   0               ;DESIRED CYLINDER REGISTER
2678  004272  000000                            .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
2679  004274  000000                            .WORD   0               ;ERROR REGISTER
2680  004276  000000                            .WORD   0               ;STATUS REGISTER
2681  004300  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
2682  004302  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
2683  004304  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
2684  004306  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
2685  004310  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
2686  004312  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
2687  004314  000000                            .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
2688  004316  000000                            .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
2689  004320  000000                            .WORD   0               ;ECC POSITION INFORMATION
2690  004322  000000                            .WORD   0               ;ECC PATTERN INFORMATION
2691  004324  000000                            .WORD   0               ;QUEUE LINK
2692  004326  000000                            .WORD   0               ;MINIMUM CYLINDER
2693  004330  000632                            .WORD   410.            ;MAXIMUM CYLINDER (DEFAULT TO RK06 MAX CYL)
2694  004332    000                             .BYTE   0               ;MINIMUM TRACK
2695  004333    002                             .BYTE   2               ;MAXIMUM TRACK
2696  004334    000                             .BYTE   0               ;MINIMUM SECTOR
2697  004335    025                             .BYTE   21.             ;MAXIMUM SECTOR
2698  004336    003                             .BYTE   3               ;READ/WRITE RATIO
2699  004337    000                             .BYTE   0               ;AUTOMATIC WRITE CHECK
2700  004340  000012                            .WORD   10.             ;CORRECTABLE READ ERROR THRESHOLD
2701  004342  000005                            .WORD   5               ;UNCORRECTABLE READ ERROR THRESHOLD
2702  004344  000005                            .WORD   5               ;SEEK ERROR THRESHOLD
2703  004346  177770  177777                    .WORD   177770,177777   ;MAXIMUM NUMBER OF COMMANDS TO DRIVE
2704  004352  177770  177777                    .WORD   177770,177777   ;MAXIMUM NUMBER OF WORDS TRANSFERRED TO DRIVE
2705  004356    000                             .BYTE   0               ;SAMPLE COMPARE FLAG
2706  004357    000                             .BYTE   0               ;ECC COMPARE FLAG
2707  004360    000                             .BYTE   0               ;INHIBIT ERROR CORRECTION
2708  004361    000                             .BYTE   0               ;DATA PATTERN USED
2709  004362    000                             .BYTE   0               ;CURRENT GENERATED DATA PATTERN
2710  004363    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED COMMAND
2711  004364  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED CYLINDER
2712  004366    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED SECTOR
2713  004367    000                             .BYTE   0               ;CURRENT RANDOMLY GENERATED TRACK
2714  004370  000000                            .WORD   0               ;CURRENT RANDOMLY GENERATED WORD COUNT
```

```
2715  004372  000000                        .WORD   0               ;CURRENT RANDOMLY GENERATED BUS ADDRESS
2716  004374     000                        .BYTE   0
2717  004375     000                        .BYTE   0               ;LAST COMMAND
2718  004376  000000                        .WORD   0               ;LAST CYLINDER
2719  004400     000                        .BYTE   0               ;LAST SECTOR
2720  004401     000                        .BYTE   0               ;LAST TRACK
2721  004402  000000                        .WORD   0               ;LAST WORD COUNT
2722  004404     000                        .BYTE   0               ;LAST DATA PATTERN USED
2723  004405     000                        .BYTE   0               ;BUFFER ALLOCATED
2724  004406     000                        .BYTE   0               ;RETRY COUNT
2725  004407     000                        .BYTE   0               ;REREAD STATUS
2726  004410  000000                        .WORD   0               ;DRIVE STATUS FLAGS
2727  004412  000000  000000                .WORD   0,0             ;NUMBER OF ORDERS
2728  004416  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS WRITTEN
2729  004424  000000  000000  000000        .WORD   0,0,0           ;NUMBER OF WORDS READ
2730  004432  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (REREAD CORRECTABLE)
2731  004434  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (OFFSET CORRECTABLE)
2732  004436  000000                        .WORD   0               ;NUMBER OF SOFT ERRORS (ECC CORRECTABLE)
2733  004440  000000                        .WORD   0               ;NUMBER OF HARD DATA ERRORS
2734  004442  000000                        .WORD   0               ;NUMBER OF SEEK INCOMPLETES
2735  004444  000000                        .WORD   0               ;NUMBER OF MISPOSITIONING ERRORS
2736  004446  000000                        .WORD   0               ;NUMBER OF ALL OTHER ERRORS
2737  004450     000                        .BYTE   0               ;ASSIGNMENT COMMAND SEQUENCE
2738  004451     000                        .BYTE   0               ;SEEK TO PREVIOUS COMMAND FLAG
2739  004452  000000                        .WORD   0               ;ERROR STATUS INFORMATION
2740  004454  000000                        .WORD   0               ;HEADER ADDRESS OF FIRST MISCOMPARE ERROR
2741  004456     000                        .BYTE   0
2742  004457     000                        .BYTE   0               ;OFFSET ADDRESS FROM HEADER ADDRESS
2743  004460  000000                        .WORD   0               ;COMPARISON LENGTH
2744  004462  000000                        .WORD   0               ;BUFFER COMPARISON LENGTH
2745  004464  007777                        .WORD   007777          ;DRIVE SERIAL NUMBER
2746  004466  000000  000000                .WORD   0,0             ;CARIAGE SERIAL NUMBER
2747  004472  000000  000000                .WORD   0,0             ;FIRST EXCLUSION AREA
2748  004476  000000  000000                .WORD   0,0             ;SECOND EXCLUSION AREA
2749  004502  000000  000000                .WORD   0,0             ;THIRD EXCLUSION AREA
2750  004506  000000  000000                .WORD   0,0             ;FOURTH EXCLUSION AREA
2751  004512  000000  000000                .WORD   0,0             ;FIFTH EXCLUSION AREA
```

```
                                .SBTTL  --- RETRY PARAMETER BLOCK ---
                                .SBTTL  PARAMETER BLOCK 10 FOR DRIVE
004516  000     PARM10: .BYTE   0               ;DRIVE NUMBER
004517  000             .BYTE   0               ;COMMAND
004520  000000          .WORD   0               ;CYLINDER ADDRESS
004522  000             .BYTE   0               ;SECTOR ADDRESS
004523  000             .BYTE   0               ;TRACK ADDRESS
004524  000             .BYTE   0               ;OFFSET VALUE
004525  000             .BYTE   0               ;BUS ADDRESS (BITS 16 AND 17)
004526  000000          .WORD   0               ;BUS ADDRESS (BITS 0 - 15)
004530  000000          .WORD   0               ;WORD COUNT (2'S COMPLEMENT)
004532  000000          .WORD   0               ;PROGRAM DRIVE STATUS INFORMATION
004534  000000          .WORD   0               ;COMMAND AND STATUS REGISTER 1
004536  000000          .WORD   0               ;COMMAND AND STATUS REGISTER 2
004540  000000          .WORD   0               ;WORD COUNT REGISTER
004542  000000          .WORD   0               ;BUS ADDRESS REGISTER
004544  000000          .WORD   0               ;DESIRED TRACK AND SECTOR REGISTER
004546  000000          .WORD   0               ;DESIRED CYLINDER REGISTER
004550  000000          .WORD   0               ;ATTENTION SUMMARY/OFFSET REGISTER
004552  000000          .WORD   0               ;ERROR REGISTER
004554  000000          .WORD   0               ;STATUS REGISTER
004556  000000          .WORD   0               ;MESSAGE LINE A STATUS BYTE 00
004560  000000          .WORD   0               ;MESSAGE LINE B STATUS BYTE 00
004562  000000          .WORD   0               ;MESSAGE LINE A STATUS BYTE 01
004564  000000          .WORD   0               ;MESSAGE LINE B STATUS BYTE 01
004566  000000          .WORD   0               ;MESSAGE LINE A STATUS BYTE 10
004570  000000          .WORD   0               ;MESSAGE LINE B STATUS BYTE 10
004572  000000          .WORD   0               ;MESSAGE LINE A STATUS BYTE 11
004574  000000          .WORD   0               ;MESSAGE LINE B STATUS BYTE 11
004576  000000          .WORD   0               ;ECC POSITION INFORMATION
004600  000000          .WORD   0               ;ECC PATTERN INFORMATION
```

```
                        .SBTTL  DATA PATTERNS AND DATA PATTERN TABLE
2785
2786
2787  004602  004642   PATTAB: PAT0
2788  004604  004742           PAT1
2789  004606  005042           PAT2
2790  004610  005142           PAT3
2791  004612  005242           PAT4
2792  004614  005342           PAT5
2793  004616  005442           PAT6
2794  004620  005542           PAT7
2795  004622  005642           PAT10
2796  004624  005742           PAT11
2797  004626  006042           PAT12
2798  004630  006142           PAT13
2799  004632  006242           PAT14
2800  004634  006342           PAT15
2801  004636  006442           PAT16
2802  004640  006542           PAT17
2803
2804  004642  165555   PAT0:   .WORD   165555
2805  004644  133333           .WORD   133333
2806  004646  165555           .WORD   165555
2807  004650  133333           .WORD   133333
2808  004652  165555           .WORD   165555
2809  004654  133333           .WORD   133333
2810  004656  165555           .WORD   165555
2811  004660  133333           .WORD   133333
2812  004662  165555           .WORD   165555
2813  004664  133333           .WORD   133333
2814  004666  165555           .WORD   165555
2815  004670  133333           .WORD   133333
2816  004672  165555           .WORD   165555
2817  004674  133333           .WORD   133333
2818  004676  165555           .WORD   165555
2819  004700  133333           .WORD   133333
2820  004702  165555           .WORD   165555
2821  004704  133333           .WORD   133333
2822  004706  165555           .WORD   165555
2823  004710  133333           .WORD   133333
2824  004712  165555           .WORD   165555
2825  004714  133333           .WORD   133333
2826  004716  165555           .WORD   165555
2827  004720  133333           .WORD   133333
2828  004722  165555           .WORD   165555
2829  004724  133333           .WORD   133333
2830  004726  165555           .WORD   165555
2831  004730  133333           .WORD   133333
2832  004732  165555           .WORD   165555
2833  004734  133333           .WORD   133333
2834  004736  165555           .WORD   165555
2835  004740  133333           .WORD   133333
2836
2837  004742  026455   PAT1:   .WORD   026455
2838  004744  151322           .WORD   151322
2839  004746  026455           .WORD   026455
2840  004750  026455           .WORD   026455
```

```
2841  004752  151322                          .WORD   151322
2842  004754  151322                          .WORD   151322
2843  004756  026455                          .WORD   026455
2844  004760  026455                          .WORD   026455
2845  004762  026455                          .WORD   026455
2846  004764  151322                          .WORD   151322
2847  004766  151322                          .WORD   151322
2848  004770  151322                          .WORD   151322
2849  004772  026455                          .WORD   026455
2850  004774  026455                          .WORD   026455
2851  004776  026455                          .WORD   026455
2852  005000  026455                          .WORD   026455
2853  005002  151322                          .WORD   151322
2854  005004  151322                          .WORD   151322
2855  005006  151322                          .WORD   151322
2856  005010  151322                          .WORD   151322
2857  005012  026455                          .WORD   026455
2858  005014  026455                          .WORD   026455
2859  005016  026455                          .WORD   026455
2860  005020  151322                          .WORD   151322
2861  005022  151322                          .WORD   151322
2862  005024  151322                          .WORD   151322
2863  005026  026455                          .WORD   026455
2864  005030  026455                          .WORD   026455
2865  005032  151322                          .WORD   151322
2866  005034  151322                          .WORD   151322
2867  005036  026455                          .WORD   026455
2868  005040  151322                          .WORD   151322
2869
2870  005042  007417                  PAT2:   .WORD   007417
2871  005044  170360                          .WORD   170360
2872  005046  007417                          .WORD   007417
2873  005050  007417                          .WORD   007417
2874  005052  170360                          .WORD   170360
2875  005054  170360                          .WORD   170360
2876  005056  007417                          .WORD   007417
2877  005060  007417                          .WORD   007417
2878  005062  007417                          .WORD   007417
2879  005064  170360                          .WORD   170360
2880  005066  170360                          .WORD   170360
2881  005070  170360                          .WORD   170360
2882  005072  007417                          .WORD   007417
2883  005074  007417                          .WORD   007417
2884  005076  007417                          .WORD   007417
2885  005100  007417                          .WORD   007417
2886  005102  170360                          .WORD   170360
2887  005104  170360                          .WORD   170360
2888  005106  170360                          .WORD   170360
2889  005110  170360                          .WORD   170360
2890  005112  007417                          .WORD   007417
2891  005114  007417                          .WORD   007417
2892  005116  007417                          .WORD   007417
2893  005120  170360                          .WORD   170360
2894  005122  170360                          .WORD   170360
2895  005124  170360                          .WORD   170360
2896  005126  007417                          .WORD   007417
```

```
2897   005130   007417                          .WORD   007417
2898   005132   170360                          .WORD   170360
2899   005134   170360                          .WORD   170360
2900   005136   007417                          .WORD   007417
2901   005140   170360                          .WORD   170360
2902
2903   005142   052525                  PAT3:   .WORD   052525
2904   005144   125252                          .WORD   125252
2905   005146   052525                          .WORD   052525
2906   005150   052525                          .WORD   052525
2907   005152   125252                          .WORD   125252
2908   005154   125252                          .WORD   125252
2909   005156   052525                          .WORD   052525
2910   005160   052525                          .WORD   052525
2911   005162   052525                          .WORD   052525
2912   005164   125252                          .WORD   125252
2913   005166   125252                          .WORD   125252
2914   005170   125252                          .WORD   125252
2915   005172   052525                          .WORD   052525
2916   005174   052525                          .WORD   052525
2917   005176   052525                          .WORD   052525
2918   005200   052525                          .WORD   052525
2919   005202   125252                          .WORD   125252
2920   005204   125252                          .WORD   125252
2921   005206   125252                          .WORD   125252
2922   005210   125252                          .WORD   125252
2923   005212   052525                          .WORD   052525
2924   005214   052525                          .WORD   052525
2925   005216   052525                          .WORD   052525
2926   005220   125252                          .WORD   125252
2927   005222   125252                          .WORD   125252
2928   005224   125252                          .WORD   125252
2929   005226   052525                          .WORD   052525
2930   005230   052525                          .WORD   052525
2931   005232   125252                          .WORD   125252
2932   005234   125252                          .WORD   125252
2933   005236   052525                          .WORD   052525
2934   005240   125252                          .WORD   125252
2935
2936   005242   000000                  PAT4:   .WORD   000000
2937   005244   010421                          .WORD   010421
2938   005246   021042                          .WORD   021042
2939   005250   031463                          .WORD   031463
2940   005252   042104                          .WORD   042104
2941   005254   052525                          .WORD   052525
2942   005256   063146                          .WORD   063146
2943   005260   073567                          .WORD   073567
2944   005262   104210                          .WORD   104210
2945   005264   114631                          .WORD   114631
2946   005266   125252                          .WORD   125252
2947   005270   135673                          .WORD   135673
2948   005272   146314                          .WORD   146314
2949   005274   156735                          .WORD   156735
2950   005276   167356                          .WORD   167356
2951   005300   177777                          .WORD   177777
2952   005302   177777                          .WORD   177777
```

```
2953   005304   167356                          .WORD   167356
2954   005306   156735                          .WORD   156735
2955   005310   146314                          .WORD   146314
2956   005312   135673                          .WORD   135673
2957   005314   125252                          .WORD   125252
2958   005316   114631                          .WORD   114631
2959   005320   104210                          .WORD   104210
2960   005322   073567                          .WORD   073567
2961   005324   063146                          .WORD   063146
2962   005326   052525                          .WORD   052525
2963   005330   042104                          .WORD   042104
2964   005332   031463                          .WORD   031463
2965   005334   021042                          .WORD   021042
2966   005336   010421                          .WORD   010421
2967   005340   000000                          .WORD   000000
2968
2969   005342   000000                  PAT5:   .WORD   000000
2970   005344   177777                          .WORD   177777
2971   005346   000000                          .WORD   000000
2972   005350   000000                          .WORD   000000
2973   005352   177777                          .WORD   177777
2974   005354   177777                          .WORD   177777
2975   005356   000000                          .WORD   000000
2976   005360   000000                          .WORD   000000
2977   005362   000000                          .WORD   000000
2978   005364   177777                          .WORD   177777
2979   005366   177777                          .WORD   177777
2980   005370   177777                          .WORD   177777
2981   005372   000000                          .WORD   000000
2982   005374   000000                          .WORD   000000
2983   005376   000000                          .WORD   000000
2984   005400   000000                          .WORD   000000
2985   005402   177777                          .WORD   177777
2986   005404   177777                          .WORD   177777
2987   005406   177777                          .WORD   177777
2988   005410   177777                          .WORD   177777
2989   005412   000000                          .WORD   000000
2990   005414   000000                          .WORD   000000
2991   005416   000000                          .WORD   000000
2992   005420   177777                          .WORD   177777
2993   005422   177777                          .WORD   177777
2994   005424   177777                          .WORD   177777
2995   005426   000000                          .WORD   000000
2996   005430   000000                          .WORD   000000
2997   005432   177777                          .WORD   177777
2998   005434   177777                          .WORD   177777
2999   005436   000000                          .WORD   000000
3000   005440   177777                          .WORD   177777
3001
3002   005442   000001                  PAT6:   .WORD   000001
3003   005444   000003                          .WORD   000003
3004   005446   000007                          .WORD   000007
3005   005450   000017                          .WORD   000017
3006   005452   000037                          .WORD   000037
3007   005454   000077                          .WORD   000077
3008   005456   000177                          .WORD   000177
```

```
3009  005460  000377                    .WORD   000377
3010  005462  000777                    .WORD   000777
3011  005464  001777                    .WORD   001777
3012  005466  003777                    .WORD   003777
3013  005470  007777                    .WORD   007777
3014  005472  017777                    .WORD   017777
3015  005474  037777                    .WORD   037777
3016  005476  077777                    .WORD   077777
3017  005500  177777                    .WORD   177777
3018  005502  177776                    .WORD   177776
3019  005504  177774                    .WORD   177774
3020  005506  177770                    .WORD   177770
3021  005510  177760                    .WORD   177760
3022  005512  177740                    .WORD   177740
3023  005514  177700                    .WORD   177700
3024  005516  177600                    .WORD   177600
3025  005520  177400                    .WORD   177400
3026  005522  177000                    .WORD   177000
3027  005524  176000                    .WORD   176000
3028  005526  174000                    .WORD   174000
3029  005530  170000                    .WORD   170000
3030  005532  160000                    .WORD   160000
3031  005534  140000                    .WORD   140000
3032  005536  100000                    .WORD   100000
3033  005540  000000                    .WORD   000000
3034
3035  005542  000001              PAT7:   .WORD   000001
3036  005544  000002                    .WORD   000002
3037  005546  000004                    .WORD   000004
3038  005550  000010                    .WORD   000010
3039  005552  000020                    .WORD   000020
3040  005554  000040                    .WORD   000040
3041  005556  000100                    .WORD   000100
3042  005560  000200                    .WORD   000200
3043  005562  000400                    .WORD   000400
3044  005564  001000                    .WORD   001000
3045  005566  002000                    .WORD   002000
3046  005570  004000                    .WORD   004000
3047  005572  010000                    .WORD   010000
3048  005574  020000                    .WORD   020000
3049  005576  040000                    .WORD   040000
3050  005600  100000                    .WORD   100000
3051  005602  100000                    .WORD   100000
3052  005604  040000                    .WORD   040000
3053  005606  020000                    .WORD   020000
3054  005610  010000                    .WORD   010000
3055  005612  004000                    .WORD   004000
3056  005614  002000                    .WORD   002000
3057  005616  001000                    .WORD   001000
3058  005620  000400                    .WORD   000400
3059  005622  000200                    .WORD   000200
3060  005624  000100                    .WORD   000100
3061  005626  000040                    .WORD   000040
3062  005630  000020                    .WORD   000020
3063  005632  000010                    .WORD   000010
3064  005634  000004                    .WORD   000004
```

```
3065  005636  000002                        .WORD   000002
3066  005640  000001                        .WORD   000001
3067
3068  005642  177776              PAT10:    .WORD   177776
3069  005644  177775                        .WORD   177775
3070  005646  177773                        .WORD   177773
3071  005650  177767                        .WORD   177767
3072  005652  177757                        .WORD   177757
3073  005654  177737                        .WORD   177737
3074  005656  177677                        .WORD   177677
3075  005660  177577                        .WORD   177577
3076  005662  177377                        .WORD   177377
3077  005664  176777                        .WORD   176777
3078  005666  175777                        .WORD   175777
3079  005670  173777                        .WORD   173777
3080  005672  167777                        .WORD   167777
3081  005674  157777                        .WORD   157777
3082  005676  137777                        .WORD   137777
3083  005700  077777                        .WORD   077777
3084  005702  077777                        .WORD   077777
3085  005704  137777                        .WORD   137777
3086  005706  157777                        .WORD   157777
3087  005710  167777                        .WORD   167777
3088  005712  173777                        .WORD   173777
3089  005714  175777                        .WORD   175777
3090  005716  176777                        .WORD   176777
3091  005720  177377                        .WORD   177377
3092  005722  177577                        .WORD   177577
3093  005724  177677                        .WORD   177677
3094  005726  177737                        .WORD   177737
3095  005730  177757                        .WORD   177757
3096  005732  177767                        .WORD   177767
3097  005734  177773                        .WORD   177773
3098  005736  177775                        .WORD   177775
3099  005740  177776                        .WORD   177776
3100
3101  005742  172666              PAT11:    .WORD   172666
3102  005744  155555                        .WORD   155555
3103  005746  172666                        .WORD   172666
3104  005750  155555                        .WORD   155555
3105  005752  172666                        .WORD   172666
3106  005754  155555                        .WORD   155555
3107  005756  172666                        .WORD   172666
3108  005760  155555                        .WORD   155555
3109  005762  172666                        .WORD   172666
3110  005764  155555                        .WORD   155555
3111  005766  172666                        .WORD   172666
3112  005770  155555                        .WORD   155555
3113  005772  172666                        .WORD   172666
3114  005774  155555                        .WORD   155555
3115  005776  172666                        .WORD   172666
3116  006000  155555                        .WORD   155555
3117  006002  172666                        .WORD   172666
3118  006004  155555                        .WORD   155555
3119  006006  172666                        .WORD   172666
3120  006010  155555                        .WORD   155555
```

```
3121   006012  172666                          .WORD   172666
3122   006014  155555                          .WORD   155555
3123   006016  172666                          .WORD   172666
3124   006020  155555                          .WORD   155555
3125   006022  172666                          .WORD   172666
3126   006024  155555                          .WORD   155555
3127   006026  172666                          .WORD   172666
3128   006030  155555                          .WORD   155555
3129   006032  172666                          .WORD   172666
3130   006034  155555                          .WORD   155555
3131   006036  172666                          .WORD   172666
3132   006040  155555                  .       .WORD   155555
3133
3134   006042  153333                  PAT12:  .WORD   153333
3135   006044  066667                          .WORD   066667
3136   006046  153333                          .WORD   153333
3137   006050  066667                          .WORD   066667
3138   006052  153333                          .WORD   153333
3139   006054  066667                          .WORD   066667
3140   006056  153333                          .WORD   153333
3141   006060  066667                          .WORD   066667
3142   006062  153333                          .WORD   153333
3143   006064  066667                          .WORD   066667
3144   006066  153333                          .WORD   153333
3145   006070  066667                          .WORD   066667
3146   006072  153333                          .WORD   153333
3147   006074  066667                          .WORD   066667
3148   006076  153333                          .WORD   153333
3149   006100  066667                          .WORD   066667
3150   006102  153333                          .WORD   153333
3151   006104  066667                          .WORD   066667
3152   006106  153333                          .WORD   153333
3153   006110  066667                          .WORD   066667
3154   006112  153333                          .WORD   153333
3155   006114  066667                          .WORD   066667
3156   006116  153333                          .WORD   153333
3157   006120  066667                          .WORD   066667
3158   006122  153333                          .WORD   153333
3159   006124  066667                          .WORD   066667
3160   006126  153333                          .WORD   153333
3161   006130  066667                          .WORD   066667
3162   006132  153333                          .WORD   153333
3163   006134  066667                          .WORD   066667
3164   006136  153333                          .WORD   153333
3165   006140  066667                          .WORD   066667
3166
3167   006142  000000                  PAT13:  .WORD   000000
3168   006144  177777                          .WORD   177777
3169   006146  177777                          .WORD   177777
3170   006150  177777                          .WORD   177777
3171   006152  177777                          .WORD   177777
3172   006154  177777                          .WORD   177777
3173   006156  177777                          .WORD   177777
3174   006160  177777                          .WORD   177777
3175   006162  177777                          .WORD   177777
3176   006164  177777                          .WORD   177777
```

```
3177  006166  177777                    .WORD   177777
3178  006170  177777                    .WORD   177777
3179  006172  177777                    .WORD   177777
3180  006174  177777                    .WORD   177777
3181  006176  177777                    .WORD   177777
3182  006200  177777                    .WORD   177777
3183  006202  177777                    .WORD   177777
3184  006204  177777                    .WORD   177777
3185  006206  177777                    .WORD   177777
3186  006210  177777                    .WORD   177777
3187  006212  177777                    .WORD   177777
3188  006214  177777                    .WORD   177777
3189  006216  177777                    .WORD   177777
3190  006220  177777                    .WORD   177777
3191  006222  177777                    .WORD   177777
3192  006224  177777                    .WORD   177777
3193  006226  177777                    .WORD   177777
3194  006230  177777                    .WORD   177777
3195  006232  177777                    .WORD   177777
3196  006234  177777                    .WORD   177777
3197  006236  177777                    .WORD   177777
3198  006240  177777                    .WORD   177777
3199
3200  006242  177777            PAT14:  .WORD   177777
3201  006244  000000                    .WORD   000000
3202  006246  000000                    .WORD   000000
3203  006250  000000                    .WORD   000000
3204  006252  000000                    .WORD   000000
3205  006254  000000                    .WORD   000000
3206  006256  000000                    .WORD   000000
3207  006260  000000                    .WORD   000000
3208  006262  000000                    .WORD   000000
3209  006264  000000                    .WORD   000000
3210  006266  000000                    .WORD   000000
3211  006270  000000                    .WORD   000000
3212  006272  000000                    .WORD   000000
3213  006274  000000                    .WORD   000000
3214  006276  000000                    .WORD   000000
3215  006300  000000                    .WORD   000000
3216  006302  000000                    .WORD   000000
3217  006304  000000                    .WORD   000000
3218  006306  000000                    .WORD   000000
3219  006310  000000                    .WORD   000000
3220  006312  000000                    .WORD   000000
3221  006314  000000                    .WORD   000000
3222  006316  000000                    .WORD   000000
3223  006320  000000                    .WORD   000000
3224  006322  000000                    .WORD   000000
3225  006324  000000                    .WORD   000000
3226  006326  000000                    .WORD   000000
3227  006330  000000                    .WORD   000000
3228  006332  000000                    .WORD   000000
3229  006334  000000                    .WORD   000000
3230  006336  000000                    .WORD   000000
3231  006340  000000                    .WORD   000000
3232
```

```
 3233  006342  172304          PAT15:  .WORD   172304
 3234  006344  172304                  .WORD   172304
 3235  006346  172304                  .WORD   172304
 3236  006350  172304                  .WORD   172304
 3237  006352  172304                  .WORD   172304
 3238  006354  172304                  .WORD   172304
 3239  006356  172304                  .WORD   172304
 3240  006360  172304                  .WORD   172304
 3241  006362  172304                  .WORD   172304
 3242  006364  172304                  .WORD   172304
 3243  006366  172304                  .WORD   172304
 3244  006370  172304                  .WORD   172304
 3245  006372  172304                  .WORD   172304
 3246  006374  172304                  .WORD   172304
 3247  006376  172304                  .WORD   172304
 3248  006400  172304                  .WORD   172304
 3249  006402  172304                  .WORD   172304
 3250  006404  172304                  .WORD   172304
 3251  006406  172304                  .WORD   172304
 3252  006410  172304                  .WORD   172304
 3253  006412  172304                  .WORD   172304
 3254  006414  172304                  .WORD   172304
 3255  006416  172304                  .WORD   172304
 3256  006420  172304                  .WORD   172304
 3257  006422  172304                  .WORD   172304
 3258  006424  172304                  .WORD   172304
 3259  006426  172304                  .WORD   172304
 3260  006430  172304                  .WORD   172304
 3261  006432  172304                  .WORD   172304
 3262  006434  172304                  .WORD   172304
 3263  006436  172304                  .WORD   172304
 3264  006440  172304                  .WORD   172304
 3265
 3266  006442  070627          PAT16:  .WORD   070627
 3267  006444  113431                  .WORD   113431
 3268  006446  014561                  .WORD   014561
 3269  006450  070627                  .WORD   070627
 3270  006452  113431                  .WORD   113431
 3271  006454  014561                  .WORD   014561
 3272  006456  070627                  .WORD   070627
 3273  006460  113431                  .WORD   113431
 3274  006462  014561                  .WORD   014561
 3275  006464  070627                  .WORD   070627
 3276  006466  113431                  .WORD   113431
 3277  006470  014561                  .WORD   014561
 3278  006472  070627                  .WORD   070627
 3279  006474  113431                  .WORD   113431
 3280  006476  014561                  .WORD   014561
 3281  006500  070627                  .WORD   070627
 3282  006502  113431                  .WORD   113431
 3283  006504  014561                  .WORD   014561
 3284  006506  070627                  .WORD   070627
 3285  006510  113431                  .WORD   113431
 3286  006512  014561                  .WORD   014561
 3287  006514  070627                  .WORD   070627
 3288  006516  113431                  .WORD   113431
```

```
3289  006520  014561                        .WORD   014561
3290  006522  070627                        .WORD   070627
3291  006524  113431                        .WORD   113431
3292  006526  014561                        .WORD   014561
3293  006530  070627                        .WORD   070627
3294  006532  113431                        .WORD   113431
3295  006534  014561                        .WORD   014561
3296  006536  070627                        .WORD   070627
3297  006540  113431                        .WORD   113431
3298
3299  006542  133467              PAT17:    .WORD   133467
3300  006544  133467                        .WORD   133467
3301  006546  133467                        .WORD   133467
3302  006550  133467                        .WORD   133467
3303  006552  133467                        .WORD   133467
3304  006554  133467                        .WORD   133467
3305  006556  133467                        .WORD   133467
3306  006560  133467                        .WORD   133467
3307  006562  133467                        .WORD   133467
3308  006564  133467                        .WORD   133467
3309  006566  133467                        .WORD   133467
3310  006570  133467                        .WORD   133467
3311  006572  133467                        .WORD   133467
3312  006574  133467                        .WORD   133467
3313  006576  133467                        .WORD   133467
3314  006600  133467                        .WORD   133467
3315  006602  133467                        .WORD   133467
3316  006604  133467                        .WORD   133467
3317  006606  133467                        .WORD   133467
3318  006610  133467                        .WORD   133467
3319  006612  133467                        .WORD   133467
3320  006614  133467                        .WORD   133467
3321  006616  133467                        .WORD   133467
3322  006620  133467                        .WORD   133467
3323  006622  133467                        .WORD   133467
3324  006624  133467                        .WORD   133467
3325  006626  133467                        .WORD   133467
3326  006630  133467                        .WORD   133467
3327  006632  133467                        .WORD   133467
3328  006634  133467                        .WORD   133467
3329  006636  133467                        .WORD   133467
3330  006640  133467                        .WORD   133467
```

```
3331                                            .SBTTL  PROGRAM SETUP
3332
3333  006642  112737  000001  001361  PARSRT: MOVB    #1,FLAG         ;SET FLAG FOR PARAMETER ENTRY
3334  006650  000412                          BR      PRGSRT          ;START PROGRAM
3335
3336  006652  112737  177777  001361  RESTRT: MOVB    #-1,FLAG        ;SET FLAG FOR RETRY
3337  006660  000406                          BR      PRGSRT          ;START PROGRAM
3338
3339  006662  105037  001361          START1: CLRB    FLAG            ;SET FOR ENTRY INPUT START
3340  006666  000403                          BR      PRGSRT          ;START PROGRAM
3341
3342  006670  112737  000002  001361  START:  MOVB    #2,FLAG         ;SET FOR AUTO SIZE PGM SIZE
3343
3344  006676  012737  000340  177776  PRGSRT: MOV     #PR7,PS         ;LOCK OUT ALL INTERRUPTS
3345                                  .SBTTL  INITIALIZE THE COMMON TAGS
3346                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
3347  006704  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
3348  006710  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
3349  006712  022706  001140                  CMP     #SWR,R6 ;;DONE?
3350  006716  001374                          BNE     .-6             ;;LOOP BACK IF NO
3351  006720  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
3352                                  ;;INITIALIZE A FEW VECTORS
3353  006724  012737  055664  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
3354  006732  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
3355  006740  012737  054552  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
3356  006746  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
3357                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
3358                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
3359  006754  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
3360  006760  012737  007014  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
3361  006766  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
3362  006774  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
3363  007002  022777  177777  172130          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
3364  007010  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
3365                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
3366  007012  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
3367  007014  012716  007022          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
3368  007020  000002                          RTI
3369  007022  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
3370  007030  012737  000174  001142          MOV     #DISPREG,DISPLAY
3371  007036  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
3372
3373  007042  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
3374  007046  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
3375  007054  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
3376  007056  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
3377  007064                          67$:
3378                                  .SBTTL  TYPE PROGRAM NAME
3379                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
3380  007064  005227  177777                  INC     #-1             ;;FIRST TIME?
3381  007070  001040                          BNE     68$             ;;BRANCH IF NO
3382  007072  104401  007100                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3383  007076  000435                          BR      68$             ;;GET OVER THE ASCIZ
3384                                  ;;69$:  .ASCIZ  <CRLF>"RK611/RK06-RK07 PERFORMANCE EXERCISER: MAINDEC DZR6P-C"<CRLF>
3385  007172                          68$:
3386  007172  000005                  PWRSRT: RESET                   ;RESET SYSTEM
```

```
3387  007174  012737  007246  000004          MOV     #20$,ERRVEC     ;SET VECTOR FOR MEMORY PARITY CHECK
3388  007202  012737  000340  000006          MOV     #PR7,ERRVEC+2
3389  007210  012703  172100                  MOV     #MEMBAS,R3      ;LOAD REGISTER TO DTERMIN IF
3390                                                                  ;  MEMORY CHECK ENABLE ABAILIABLE
3391  007214  012704  000014                  MOV     #12.,R4         ;LOAD COUNT
3392  007220  012723  000001          16$:    MOV     #PAR.EN,(R3)+   ;ENABLE MEMORY CHECK VECTORY
3393  007224  012737  052472  000114          MOV     #MEMERR,MEMVEC  ;LOAD MEMORY CHECK VECTOR
3394  007232  012737  000340  000116          MOV     #PR7,MEMVEC+2
3395  007240  005304                          DEC     R4              ;CHECK IF FINISHED
3396  007242  001366                          BNE     16$             ;NO, SET UP NEXT MEMORY PARITY MODULE
3397  007244  000401                          BR      22$             ;RESTORE TRAP VECTOR
3398
3399  007246  022626          20$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
3400  007250  012737  000006  000004  22$:    MOV     #ERRVEC+2,ERRVEC ;RESTORE TRAP CATCHER
3401  007256  005037  000006                  CLR     ERRVEC+2
3402  007262  004737  054454          START$: JSR     PC,$SIZE        ;GET MAXIMUM MEMORY ADDRESS
3403  007266  012737  000001  001670          MOV     #1,FBLKC        ;INITIALIZE FREE BLOCK COUNT
3404  007274  012703  001672                  MOV     #FBLKT,R3       ;STORE FREE BLOCK TABLE ADDRESS
3405  007300  012704  000022                  MOV     #18.,R4         ;LOAD COUNT FOR FREE BLOCK CLEAR
3406  007304  005023                  2$:     CLR     (R3)+           ;CLEAR FREE BLOCK TABLE
3407  007306  005304                          DEC     R4              ;DECREMENT COUNT
3408  007310  001375                          BNE     2$              ;CHECK IF FINISHED
3409  007312  012737  066212  001672          MOV     #BUFADD,FBLKT   ;LOAD FIRST ENTRY IN FREE BLOCK TABLE
3410  007320  105037  001644                  CLRB    RTYSCN          ;CLEAR RETRY SECTOR COUNT
3411  007324  005037  001566                  CLR     ERRPRO          ;CLEAR ADDRESS OF PARAMETER BLOCK
3412                                                                  ;  PROCESSING ERROR
3413  007330  005037  001462                  CLR     E.CONT          ;CLEAR CONTROLLER ERROR FLAGS
3414  007334  005037  001564                  CLR     ERCONT          ;CLEAR RECOVERABLE CONTROLLER ERROR FLAGS
3415  007340  005037  001626                  CLR     DLTCNT          ;CLEAR DATA LATE COUNT
3416  007344  005037  001630                  CLR     CNTCNT          ;CLEAR OTHER ERRORS COUNT
3417  007350  105037  001633                  CLRB    STATIS          ;CLEAR INTERVAL STATISTICS FLAG
3418  007354  112737  177777  001513          MOVB    #-1,O.OVER      ;SET ALL DRIVES FOR IMPLIED SEEKS
3419  007362  012703  001314                  MOV     #AVAILQ,R3      ;STORE QUEUE BASE ADDRESS
3420  007366  012704  000010                  MOV     #8.,R4          ;LOAD QUEUE COUNT
3421  007372  005023                  3$:     CLR     (R3)+           ;CLEAR QUEUE HEADS AND TAILS
3422  007374  005304                          DEC     R4              ;DECREMENT COUNT
3423  007376  001375                          BNE     3$              ;CHECK IF FINISHED
3424  007400  012704  001544                  MOV     #W.DRV,R4       ;LOAD ADDRESS OF TIME OUT COUNTS
3425  007404  012703  000010                  MOV     #8.,R3          ;LOAD COUNT
3426  007410  005024                  4$:     CLR     (R4)+           ;CLEAR DRIVE TIME OUT COUNTS
3427  007412  005303                          DEC     R3              ;DECREMENT COUNT
3428  007414  001375                          BNE     4$              ;CHECK IF FINISHED
3429  007416  004737  043372                  JSR     PC,CHKCLK       ;CHECK IF CLOCK IS PRESENT
3430  007422  105037  001512                  CLRB    W.TIME          ;CLEAR DRIVES BEING TIMED
3431  007426  013737  001470  001466          MOV     W.MILI,W.MTIM   ;INITIALIZE MILI-SECOND TIMER
3432  007434  005037  001464                  CLR     O.WAIT          ;CLEAR PARAMETER BLOCK FOR DRIVE
3433                                                                  ;  WAITING COMMAND COMPLETION
3434  007440  105037  001506                  CLRB    I.ISRL          ;RESET RELEASE OR CONTROLLER CLEAR ISSUED
3435  007444  105737  001361                  TSTB    FLAG            ;CHECK IF RESTART
3436  007450  100072                          BPL     RESTO           ;NO, CONTINUE
3437
3438  007452  013737  054550  001674          MOV     $LSTAD,FBLKT+2  ;LOAD MAX. ADD IF FREE BLOCK TABLE
3439  007460  105737  001365                  TSTB    OVLYLD          ;CHECK IF OVERLAY LOADER
3440  007464  001012                          BNE     20$             ;YES, LOAD RK06 VECTOR ADDRESS
3441  007466  105737  000041                  TSTB    41              ;SEE WHO LOADED THE PROBLEM
3442  007472  001004                          BNE     10$             ;BRANCH IF XXDP
```

```
3443  007474  162737  000300  001674          SUB     #96.*2.,FBLKT+2 ;SUBTRACT ABS LOADER SIZE
3444  007502  000403                          BR      20$             ;LOAD VECTOR ADDRESS
3445
3446  007504  162737  004004  001674  10$:    SUB     #1026.*2.,FBLKT+2 ;SUBTRACT XXDP LOADER
3447  007512  013704  001450          20$:    MOV     RKVEC,R4        ;STORE VECTOR ADDRESS
3448  007516  012724  044266                  MOV     #I.INTR,(R4)+   ;LOAD RK06 VECTOR ADDRESS
3449  007522  013714  001452                  MOV     RKPRI,(R4)      ;  AND INTERRUPT PRIORITY
3450  007526  012703  001524                  MOV     #PBLKT,R3       ;LOAD PARAMETER BLOCK TABLE ADDRESS
3451  007532  012704  000011                  MOV     #9.,R4          ;LOAD COUNT
3452
3453  007536  005304                  21$:    DEC     R4              ;DECREMENT COUNT AND CHECK IF DONE
3454  007540  001434                          BEQ     30$             ;YES, SET UP TTY INTERRUPT
3455  007542  012305                          MOV     (R3)+,R5        ;GET PARAMETER BLOCK ADDRESS
3456  007544  105065  000145                  CLRB    P.BUFF(R5)      ;CLEAR BUFFER ALLOCATED
3457  007550  005065  000212                  CLR     P.ERR(R5)       ;CLEAR ERROR FLAGS AND RETRY COUNTS
3458  007554  005065  000150                  CLR     P.DSTT(R5)
3459  007560  005065  000146                  CLR     P.RECT(R5)
3460
3461
3462                                  ;       RESET THE FOLLOWING DRIVER FLAGS
3463                                  ;           DRIVE POSITIONING
3464                                  ;           DRIVE POSITIONING FOR DATA TRANSFER
3465                                  ;           DRIVE SEIZED
3466                                  ;           DRIVE UNLOADED DUE TO ERROR
3467                                  ;           PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
3468  007564  042765  070006  000014            BIC     #DRVPOS!DRVPDT!DRVSZD!E.UNLD!Q.INIT,P.PRST(R5)
3469
3470  007572  032765  000001  000014            BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
3471  007600  001756                          BEQ     21$             ;NO, GET NEXT PARAMETER BLOCK
3472  007602  105765  000210                  TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
3473  007606  001003                          BNE     22$             ;YES, RESTART ASSIGNMENT SEQUENCE
3474  007610  004737  013434                  JSR     PC,TEST1        ;ISSUE A START SPINDLE IF NECESSARY
3475  007614  000750                          BR      21$             ;GET NEXT PARAMETER BLOCK
3476
3477  007616  142765  177757  000210  22$:    BICB    #^C<BIT4>,P.ASSN(R5) ;KEEP WRITE PACK BIT
3478  007624  004737  013360                  JSR     PC,TEST         ;INITIATE DRIVE ASSIGNMENT SEQUENCE
3479  007630  000742                          BR      21$             ;GET NEXT PARAMETER BLOCK
3480
3481  007632  000137  011252          30$:    JMP     REST2           ;SET TTY INTERRUPT
3482
3483  007636  012737  176543  055126  RESTO:  MOV     #176543,$HINUM  ;INITIALIZE RANDOM NUMBER GENERATOR
3484  007644  012737  123456  055130          MOV     #123456,$LONUM
3485  007652  005037  001372                  CLR     SECOND          ;INITIALIZE INTERNAL CLOCK
3486  007656  005037  001370                  CLR     MINUTE
3487  007662  005037  001366                  CLR     HOUR
3488  007666  012703  001524                  MOV     #PBLKT,R3       ;LOAD PARAMETER BLOCK TABLE ADDRESS
3489  007672  012704  000010                  MOV     #8.,R4          ;LOAD COUNT
3490  007676  012305                  4$:     MOV     (R3)+,R5        ;GET PARAMETER BLOCK ADDRESS
3491  007700  005065  000014                  CLR     P.PRST(R5)      ;CLEAR RK06 PROGRAM DEVICE STATUS REG.
3492  007704  005065  000212                  CLR     P.ERR(R5)       ;CLEAR DRIVE ERROR FLAGS
3493  007710  105065  000210                  CLRB    P.ASSN(R5)      ;CLEAR DRIVE ASSIGNMENT CODE
3494  007714  010500                          MOV     R5,R0           ;STORE R0 FOR TRACK EXCLUSIONS
3495  007716  062700  000232                  ADD     #P.EXAR,R0      ;DETERMINE TRACK EXCLUSION BLOCK
3496  007722  012701  000012                  MOV     #10.,R1         ;LOAD COUNT FOR CLEARING TRACK EXCLUSION
3497                                                                  ;  BLOCK
3498  007726  005020                  5$:     CLR     (R0)+           ;CLEAR TRACK EXCLUSION ENTRY
```

```
3499  007730  005301                        DEC     R1              ;DECREMENT COUNT
3500  007732  001375                        BNE     5$              ;CHECK IF WHOLE BLOCK CLEARED
3501  007734  005304                        DEC     R4              ;DECREMENT COUNT
3502  007736  001357                        BNE     4$              ;TEST IF DONE
3503  007740  122737  000001  001361        CMPB    #1,FLAG         ;CHECK IF PARAMETER START
3504  007746  001402                        BEQ     10$             ;YES, ASK FOR PARAMETERS
3505  007750  000137  011102                JMP     S1$             ;USE DEFAULT VALUES
3506
3507  007754  104401  057161        10$:    TYPE    ,SYS000         ;TYPE "CPU ID"
3508  007760  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3509  007764  010014                        11$                     ;COMMA DETECTED
3510  007766  010032                        13$                     ;CARRIAGE RETURN DETECTED
3511  007770  011102                        S1$                     ;CONTROL Z <↑Z> DETECTED
3512  007772  007754                        10$                     ;CONTROL C <↑C> DETECTED
3513  007774  010346                        MOV     R3,-(SP)        ;LOAD BUFFER ADDRESS ON STACK
3514  007776  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
3515  010002  010014                        11$                     ;ERROR RETURN
3516  010004  012604                        MOV     (SP)+,R4        ;STORE INPUT VALUE
3517  010006  022704  000400                CMP     #400,R4         ;CHECK IF 0-377
3518  010012  101010                        BHI     14$             ;YES, LOAD CPU ID
3519  010014                  11$:
3520  010014  010337  010022                MOV     R3,64$          ;LOAD BUFFER ADDRESS
3521  010020  104401                        TYPE                    ;TYPE RECEIVED INPUT
3522  010022  000000                64$:    .WORD   0               ;BUFFER ADDRESS
3523  010024  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
3524  010030  000751                        BR      10$             ;TRY AGAIN
3525
3526  010032  005004                13$:    CLR     R4              ;CLEAR CPU ID (R4)
3527  010034  110437  001364        14$:    MOVB    R4,CPUID        ;LOAD CPU ID
3528
3529  010040  013737  054550  001674  15$:  MOV     $LSTAD,FBLKT+2  ;LOAD LAST ADDRESS IN FIRST
3530                                                                ;   FREE BLOCK ENTRY
3531  010046  104401  057171                TYPE    ,SYS001         ;TYPE "OVERLAY LOADER"
3532  010052  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3533  010056  010126                        18$                     ;COMMA DETECTED
3534  010060  010074                        16$                     ;CARRIAGE RETURN DETECTED
3535  010062  011114                        S2$                     ;CONTROL Z <↑Z> DETECTED
3536  010064  007754                        10$                     ;CONTROL C <↑C> DETECTED
3537  010066  122713  000131                CMPB    #'Y,(R3)        ;CHECK IF YES
3538  010072  001424                        BEQ     19$             ;YES, INDICATE THAT LOADER HAS BEEN OVERLAYED
3539  010074  105037  001365        16$:    CLRB    OVLYLD          ;INDICATE THAT LOADER HAS NOT BEEN OVERLAYED
3540  010100  105737  000041                TSTB    41              ;SEE WHO LOADED THE PROGRAM
3541  010104  001004                        BNE     17$             ;BRANCH IF XXDP
3542  010106  162737  000300  001674        SUB     #96.*2.,FBLKT+2 ;SUBTRACT ABS LOADER SIZE
3543  010114  000416                        BR      20$             ;ASK FOR MAXIMUM TRANSFER SIZE
3544
3545  010116  162737  006000  001674  17$:  SUB     #1536.*2.,FBLKT+2 ;SUBTRACT XXDP LOADER SIZE
3546  010124  000412                        BR      20$             ;ASK FOR TRANSFER SIZE
3547
3548  010126                  18$:
3549  010126  010337  010134                MOV     R3,65$          ;LOAD BUFFER ADDRESS
3550  010132  104401                        TYPE                    ;TYPE RECEIVED INPUT
3551  010134  000000                65$:    .WORD   0               ;BUFFER ADDRESS
3552  010136  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
3553  010142  000736                        BR      15$             ;TRY AGAIN
3554
```

```
3555  010144  112737  177777  001365  19$:    MOVB    #-1,OVLYLD      ;INDICATE THAT LOADER HAS BEEN OVERLAYED
3556
3557  010152  104401  057211          20$:    TYPE    ,SYS002         ;TYPE "MAXIMUM TRANSFER"
3558  010156  013746  001674                  MOV     FBLKT+2,-(SP)   ;STORE LAST USABLE MEMORY LOCATION
3559  010162  163716  001672                  SUB     FBLKT,(SP)      ;CALCULATE MEMORY LEFT
3560  010166  022716  027000                  CMP     #256.*23.*2.,(SP) ;CHECK IF GREATER THAN OF EQUAL TO 5888
3561  010172  101402                          BLOS    21$             ;YES, USE 5888
3562  010174  006216                          ASR     (SP)            ;DETERMINE NUMBER OF WORDS
3563  010176  000402                          BR      22$             ;TYPE VALUE
3564
3565  010200  012716  013400          21$:    MOV     #256.*23.,(SP)  ;LOAD 5888
3566
3567  010204  012637  001570          22$:    MOV     (SP)+,TMPCNT    ;SAVE TRANSFER COUNT
3568  010210  013746  001570                  MOV     TMPCNT,-(SP)    ;SAVE TRANSFER COUNT FOR TYPE OUT
3569  010214  004737  052774                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
3570  010220  104401  057243                  TYPE    ,SYS003
3571  010224  004037  020340                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3572  010230  010272                          24$                     ;COMMA DETECTED
3573  010232  010310                          26$                     ;CARRIAGE RETURN DETECTED
3574  010234  011144                          53$                     ;CONTROL Z <↑Z> DETECTED
3575  010236  007754                          10$                     ;CONTROL C <↑C> DETECTED
3576  010240  010346                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS ON STACK
3577  010242  004737  054146                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
3578  010246  010270                          23$                     ;ERROR RETURN
3579  010250  012604                          MOV     (SP)+,R4        ;STORE INPUT
3580  010252  001407                          BEQ     24$             ;CHECK IF ZERO
3581  010254  020437  001570                  CMP     R4,TMPCNT       ;CHECK IF LEGAL ENTRY
3582  010260  101004                          BHI     24$             ;NO, TRY AGAIN
3583  010262  010437  001354                  MOV     R4,MAXBUF       ;LOAD MAXIMUM TRANSFER
3584  010266  000422                          BR      27$             ;GET    NO. OF SOFT COMPARES
3585
3586  010270  005726          23$:    TST     (SP)+           ;ADJUST STACK
3587  010272                  24$:
3588  010272  010337  010300                  MOV     R3,66$          ;LOAD BUFFER ADDRESS
3589  010276  104401                          TYPE                    ;TYPE RECEIVED INPUT
3590  010300  000000          66$:    .WORD   0               ;BUFFER ADDRESS
3591  010302  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
3592  010306  000721                          BR      20$             ;TRY AGAIN
3593
3594  010310  013737  001570  001354  26$:    MOV     TMPCNT,MAXBUF ;LOAD MAXIMUM TRANSFER
3595  010316  022737  013400  001354          CMP     #256.*23.,MAXBUF ;CHECK IF LESS THAN OR EQUAL TO 23 SECTORS
3596  010324  103003                          BHIS    27$             ;YES, USE CALCULATED VALUE
3597  010326  012737  013400  001354          MOV     #256.*23.,MAXBUF ;USE 23 SECTORS
3598
3599  010334  104401  057247          27$:    TYPE    ,SYS004         ;TYPE "NO. OF SOFTWARE COMPARISONS"
3600  010340  004037  020340                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3601  010344  010402                          29$                     ;COMMA DETECTED
3602  010346  010420                          30$                     ;CARRIAGE RETURN DETECTED
3603  010350  011202                          54$                     ;CONTROL Z <↑Z> DETECTED
3604  010352  007754                          10$                     ;CONTROL C <↑C> DETECTED
3605  010354  010346                          MOV     R3,-(SP)        ;STORE PARMETER
3606  010356  004737  054302                  JSR     PC,DECBIN       ;CONVERT TO BINARY
3607  010362  010402                          29$                     ;ERROR RETURN
3608  010364  023716  001354                  CMP     MAXBUF,(SP)     ;CHECK IF LEGAL INPUT
3609  010370  103403                          BLO     28$             ;TYPE VALUE AND RETRY
3610  010372  012637  001362                  MOV     (SP)+,SOFCMP    ;LOAD NUMBER OF SOFTWARE COMPARES
```

```
3611  010376  000413                        BR      31$             ;DETERMINE RK06 UNIBUS ADDRESS
3612
3613  010400  005726                28$:    TST     (SP)+           ;ADJUST STACK
3614  010402                      29$:
3615  010402  010337  010410              MOV     R3,67$          ;LOAD BUFFER ADDRESS
3616  010406  104401                      TYPE                    ;TYPE RECEIVED INPUT
3617  010410  000000                67$:  .WORD   0               ;BUFFER ADDRESS
3618  010412  104401  001164              TYPE    ,$QUES          ;TYPE QUESTION MARK
3619  010416  000746                      BR      27$             ;TRY AGAIN
3620
3621  010420  012737  000003  001362  30$:  MOV     #3,SOFCMP       ;LOAD DEFAULT VALUE
3622
3623  010426  105737  001374        31$:  TSTB    CLKFLG          ;CHECK IF CLOCK ON SYSTEM
3624  010432  001434                      BEQ     41$             ;NO, CONTINUE
3625  010434  104401  057442              TYPE    ,SYS009         ;TYPE "STATISTIC INTERVAL="
3626  010440  004037  020340              JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3627  010444  010502                      34$                     ;COMMA DETECTED
3628  010446  010520                      35$                     ;CARRIAGE RETURN DETECTED
3629  010450  011210                      5$                      ;CONTROL Z <↑Z> DETECTED
3630  010452  007754                      10$                     ;CONTROL C <↑C> DETECTED
3631  010454  010346                      MOV     R3,-(SP)        ;STORE NUMBER FOR CONVERSION
3632  010456  004737  054302              JSR     PC,DECBIN       ;CONVERT TO BINARY
3633  010462  010502                      34$                     ;ILLEGAL NUMBER
3634  010464  012604                      MOV     (SP)+,R4        ;STORE NUMBER
3635  010466  022704  000377              CMP     #255.,R4        ;CHECK IF GREATER THAN 255
3636  010472  103403                      BLO     34$             ;NO, TRY AGAIN
3637  010474  110437  001360              MOVB    R4,PERINV       ;LOAD PERFORMANCE INTERVAL
3638  010500  000411                      BR      41$             ;CONTINUE
3639
3640  010502                      34$:
3641  010502  010337  010510              MOV     R3,68$          ;LOAD BUFFER ADDRESS
3642  010506  104401                      TYPE                    ;TYPE RECEIVED INPUT
3643  010510  000000                68$:  .WORD   0               ;BUFFER ADDRESS
3644  010512  104401  001164              TYPE    ,$QUES          ;TYPE QUESTION MARK
3645  010516  000743                      BR      31$             ;TRY AGAIN
3646
3647  010520  105037  001360        35$:  CLRB    PERINV          ;SET UP FOR NO INTERVAL STATISTICS
3648
3649  010524  104401  057301        41$:  TYPE    ,SYS005         ;TYPE "RK06 BUS ADD."
3650  010530  004037  020340              JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3651  010534  010650                      44$                     ;COMMA DETECTED
3652  010536  010666                      45$                     ;CARRIAGE RETURN DETECTED
3653  010540  011214                      S6$                     ;CONTROL Z <↑Z> DETECTED
3654  010542  007754                      10$                     ;CONTROL C <↑C> DETECTED
3655  010544  010346                      MOV     R3,-(SP)        ;STORE PARAMETER
3656  010546  004737  054146              JSR     PC,OCTBIN       ;CONVERT TO BINARY
3657  010552  010650                      44$                     ;ERROR RETURN
3658  010554  012604                      MOV     (SP)+,R4        ;STORE INPUT
3659  010556  022704  160000              CMP     #160000,R4      ;CHECK IF I/O PAGE
3660  010562  101032                      BHI     44$             ;NO,TRY AGAIN
3661  010564  012737  010624  000004      MOV     #42$,ERRVEC     ;LOAD ERRVEC TO NON-EXISTENT MEMORY
3662  010572  013737  000340  000006      MOV     PR7,ERRVEC+2    ;   PRIORITY 7
3663  010600  005714                      TST     (R4)            ;CHECK FOR NON-EXISTENT MEMORY
3664  010602  012737  000000  000006      MOV     #<HALT>,ERRVEC+2 ;REINSTATE TRAP CATCHER
3665  010610  012737  000006  000004      MOV     #ERRVEC+2,ERRVEC
3666  010616  010437  001446              MOV     R4,RKBAS        ;LEGAL I/O ADDRESS, LOAD RK06 BASE
```

```
3667  010622  000424                        BR      48$             ;GET VECTOR ADDRESS
3668
3669  010624  012737  000000  000006  42$:  MOV     #<HALT>,ERRVEC+2 ;REINSTATE TRAP CATCHER
3670  010632  012737  000006  000004        MOV     #ERRVEC+2,ERRVEC
3671  010640  062706  000004                ADD     #4,SP           ;ADJUST STACK
3672  010644  104401  057347                TYPE    ,SYS007         ;TYPE "NON-EXISTENT MEMORY"
3673  010650                          44$:
3674  010650  010337  010656                MOV     R3,69$          ;LOAD BUFFER ADDRESS
3675  010654  104401                        TYPE                    ;TYPE RECEIVED INPUT
3676  010656  000000                  69$:  .WORD   0               ;BUFFER ADDRESS
3677  010660  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
3678  010664  000717                        BR      41$             ;TRY AGAIN
3679
3680  010666  012737  177440  001446  45$:  MOV     #177440,RKBAS   ;LOAD DEFAULT BUS ADDRESS
3681
3682  010674  104401  057324          48$:  TYPE    ,SYS006         ;TYPE "RK06 VEC ADD"
3683  010700  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3684  010704  010742                        50$                     ;COMMA DETECTED
3685  010706  010760                        51$                     ;CARRIAGE RETURN DETECTED
3686  010710  011222                        57$                     ;CONTROL Z <↑Z> DETECTED
3687  010712  007754                        10$                     ;CONTROL C <↑C> DETECTED
3688  010714  010346                        MOV     R3,-(SP)        ;STORE PARAMETER
3689  010716  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
3690  010722  010742                        50$                     ;ERROR RETURN
3691  010724  022716  001000                CMP     #1000,(SP)      ;CHECK IF VECTOR SPACE
3692  010730  101403                        BLOS    49$             ;NO, TRY AGAIN
3693  010732  012637  001450                MOV     (SP)+,RKVEC     ;LOAD VECTOR ADDRESS
3694  010736  000413                        BR      53$             ;GET PRIORITY
3695
3696  010740  005726                  49$:  TST     (SP)+           ;ADJUST STACK
3697  010742                          50$:
3698  010742  010337  010750                MOV     R3,70$          ;LOAD BUFFER ADDRESS
3699  010746  104401                        TYPE                    ;TYPE RECEIVED INPUT
3700  010750  000000                  70$:  .WORD   0               ;BUFFER ADDRESS
3701  010752  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
3702  010756  000746                        BR      48$             ;TRY AGAIN
3703
3704  010760  012737  000210  001450  51$:  MOV     #210,RKVEC      ;LOAD DEFAULT RK06 VECTOR
3705  010766  104401  057466          53$:  TYPE    ,SYS010         ;TYPE "RK06 PRIOR="
3706  010772  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
3707  010776  011054                        56$                     ;COMMA DETECTED
3708  011000  011072                        57$                     ;CARRIAGE RETURN DETECTED
3709  011002  011230                        58$                     ;CONTROL Z <↑Z> DETECTED
3710  011004  007754                        10$                     ;CONTROL C <↑C> DETECTED
3711  011006  010346                        MOV     R3,-(SP)        ;STORE PARAMETER
3712  011010  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
3713  011014  011054                        56$                     ;ERROR RETURN
3714  011016  022716  000007                CMP     #7,(SP)         ;CHECK IF OCTAL
3715  011022  103413                        BLO     55$
3716  011024  022716  000004                CMP     #4,(SP)
3717  011030  101010                        BHI     55$
3718  011032  006316                        ASL     (SP)            ;SHIFT 5 BITS LEFT
3719  011034  006316                        ASL     (SP)
3720  011036  006316                        ASL     (SP)
3721  011040  006316                        ASL     (SP)
3722  011042  006316                        ASL     (SP)
```

```
3723  011044  012637  001452          MOV     (SP)+,RKPRI     ;LOAD PRIORITY
3724  011050  000472                  BR      REST1           ;LOAD RK06 VECTOR AND TTY VECTOR
3725
3726  011052  005726          55$:    TST     (SP)+           ;ADJUST STACK
3727  011054                  56$:
3728  011054  010337  011062          MOV     R3,71$          ;LOAD BUFFER ADDRESS
3729  011060  104401                  TYPE                    ;TYPE RECEIVED INPUT
3730  011062  000000          71$:    .WORD   0               ;BUFFER ADDRESS
3731  011064  104401  001164          TYPE    ,$QUES          ;TYPE QUESTION MARK
3732  011070  000736                  BR      53$             ;TRY AGAIN
3733
3734  011072  012737  000240  001452 57$:    MOV     #PR5,RKPRI      ;LOAD PRIORITY
3735  011100  000456                  BR      REST1           ;LOAD RK06 VECTOR AND TTY VECTOR
3736
3737  011102  105037  001364          S1$:    CLRB    CPUID           ;LOAD DEFAULT CPU ID
3738  011106  013737  054550  001674          MOV     $LSTAD,FBLKT+2  ;LOAD LAST ADDRESS
3739  011114  105037  001365          S2$:    CLRB    OVLYLD          ;INDICATE THAT THE LOADER IS NOT OVERLAYED
3740  011120  105737  000041                  TSTB    41              ;SEE WHO LOADED PROGRAM
3741  011124  001004                  BNE     1$              ;BRANCH IF XXDP
3742  011126  162737  000300  001674          SUB     #96.*2.,FBLKT+2 ;SUBTRACT ABS LOADER SIZE
3743  011134  000403                  BR      S3$             ;GO DETERMINE MAX BUFFER SIZE
3744
3745  011136  162737  006000  001674 1$:     SUB     #1536.*2.,FBLKT+2 ;SUBTRACT XXDP LOADER SIZE
3746  011144  013746  001674          S3$:    MOV     FBLKT+2,-(SP)   ;STORE MAXIMUM ADDRESS
3747  011150  163716  001672                  SUB     FBLKT,(SP)      ;DETERMINE MEMORY AVAILIABLE
3748  011154  022716  027000                  CMP     #256.*#23.*2.,(SP) ;CHECK IF ENOUGH ROOM
3749  011160  101005                  BHI     1$              ;NO,LOAD MEMORY AVILIABLE
3750  011162  005726                  TST     (SP)+           ;ADJUST STACK
3751  011164  012737  013400  001354          MOV     #256.*#23.,MAXBUF ;LOAD MAXIMUM BUFFER SIZE
3752  011172  000403                  BR      S4$             ;LOAD NUMBER OF SOFTWARE COMPARES
3753
3754  011174  006216          1$:     ASR     (SP)            ;CHANGE TO WORDS
3755  011176  012637  001354          MOV     (SP)+,MAXBUF    ;LOAD MAXIMUM WORDS TRANSFERRED
3756  011202  012737  000003  001362 S4$:     MOV     #3,SOFCMP       ;LOAD NO. OF SOFTWARE COMPARES
3757  011210  105037  001360          S5$:    CLRB    PERINV          ;SET UP FOR NO INTERVAL STATISTICS
3758  011214  012737  177440  001446 S6$:     MOV     #177440,RKBAS   ;LOAD DEFAULT BUS ADDRESS
3759  011222  012737  000210  001450 S7$:     MOV     #210,RKVEC      ;LOAD DEFAULT VECTOR ADDRESS
3760  011230  012737  000240  001452 S8$:     MOV     #PR5,RKPRI      ;LOAD RK06 PRIORITY
3761
3762  011236  013704  001450          REST1:  MOV     RKVEC,R4        ;STORE VECTOR ADDRESS
3763  011242  012724  044266                  MOV     #I.INTR,(R4)+   ;LOAD RK06 VECTOR ADDRESS
3764  011246  013714  001452                  MOV     RKPRI,(R4)      ;  AND INTERRUPT PRIORITY
3765  011252                  REST2:
3766  011252  122737  000013  000041          CMPB    #13,@#41        ;LOAD FROM XXDP ?
3767  011260  001003                  BNE     99$             ;BRANCH IF NOT
3768  011262  104401  055730                  TYPE    ,XXDPMG
3769  011266  000000                  HALT
3770  011270                  99$:
3771  011270  004737  043464                  JSR     PC,CLKINT       ;INITIALIZE CLOCK VECTOR ADDRESS
3772  011274  104401  057374                  TYPE    ,SYSOO8         ;TYPE "READY TO BEGIN TESTING"
3773  011300  005227  177777                  INC     #-1             ;TYPE HELP MESSAGE ON FIRST PASS
3774  011304  001002                  BNE     5$
3775  011306  104401  066006                  TYPE    ,HELP
3776  011312  004737  053542          5$:     JSR     PC,$TKINT       ;SET UP TTY INTERRUPT
3777  011316  013737  001452  000036          MOV     RKPRI,TRAPVEC+2 ;ALLOW L AND P CLOCK INTERRUPTS
3778  011324  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
```

```
3779
3780  011330  122737  000002  001361        CMPB    #2,FLAG         ;SEE IF AUTO SIZE
3781  011336  001401                        BEQ     1$              ;BR IF YES
3782  011340  000406                        BR      MAIN            ;ELSE GOTO MAIN
3783  011342  012746  011356          1$:   MOV     #MAIN,-(SP)     ;SETUP STACK TO RETRUN TO MAIN
3784  011346  010446                        MOV     R4,-(SP)        ;SAVE R4
3785  011350  010546                        MOV     R5,-(SP)        ;SAVE R5
3786  011352  000137  012624                JMP     WRTEPK+10               ;WRITE ALL DRIVES ON BUS & RET TO MAIN
3787
```

```
                                .SBTTL  MAIN IDLE LOOP

011356  022706  001100          MAIN:   CMP     #STACK,SP       ;CHECK IF STACK PROBLEM
011362  001401                          BEQ     1$              ;NO, CONTINUE
011364  000000                          HALT                    ;  *** PROGRAM PROBLEM
011366  004737  051740          1$:     JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
011372  004737  044112                  JSR     PC,W.WTCH       ;CALL WATCH DOG TIMER
011376  005737  001564                  TST     ERCONT          ;CHECK IF IN SPECIAL ERROR SEQUENCE
011402  100765                          BMI     MAIN            ;YES, DO NOT PRINT INTERVAL STATISTICS OR
                                                                ;  ADD DRIVES TO TESTING SEQUENCE
011404  032777  060000  167526          BIT     #SW14!SW13,@SWR ;CHECK IF LOOP ON OPERATION OR
                                                                ;  INHIBIT PRINT OUT
011412  001022                          BNE     11$             ;YES, DO NOT PRINT OUT INTERVAL STATISTICS
011414  105737  001633                  TSTB    STATIS          ;CHECK FOR INTERVAL STATISTICS
011420  001417                          BEQ     11$             ;NO, CONTINUE SCAN
011422  005004                          CLR     R4              ;START WITH DRIVE 0
011424  016405  001524          2$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
011430  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IS IN USE
011436  001402                          BEQ     3$              ;NO, LOOK AT NEXT PARAMETER BLOCK
011440  004737  013102                  JSR     PC,STAT         ;GET STATISTICS
011444  005724                  3$:     TST     (R4)+           ;INCREMENT DRIVE INDEX
011446  022704  000020                  CMP     #20,R4          ;CHECK IF ALL DRIVES SCANNED
011452  101364                          BHI     2$              ;NO, DO NEXT DRIVE
011454  105037  001633                  CLRB    STATIS          ;RESET FLAG

011460  004037  051466          11$:    JSR     R0,Q.POP        ;GET FIRST DRIVE FROM DRIVE
011464  001314                          AVAILQ                  ;  AVAILABLE QUEUE
011466  012605                          MOV     (SP)+,R5        ;STORE PARAMETER BLOCK ADDRESS
011470  001732                          BEQ     MAIN            ;IF QUEUE EMPTY WAIT
011472  032765  002000  000014          BIT     #DRPDRV,P.PRST(R5)  ;CHECK IF DROP DRIVE
011500  001403                          BEQ     12$             ;NO, GENERATE NEW COMMAND
011502  004737  013004                  JSR     PC,DROP         ;DROP DRIVE
011506  000723                          BR      MAIN            ;GET NEXT AVAILABLE DRIVE

011510  032777  040000  167422  12$:    BIT     #SW14,@SWR      ;CHECK IF LOOP ON OPERATION
011516  001002                          BNE     15$             ;YES, DO NOT CLEAR SEEK TO PREVIOUS
                                                                ;  CYLINDER FLAG
011520  105065  000211                  CLRB    P.SEEK(R5)      ;CLEAR SEEK TO PREVIOUS CYLINDER
011524  004737  021160          15$:    JSR     PC,GENERT       ;GENERATE NEW COMMAND
011530  105765  000211                  TSTB    P.SEEK(R5)      ;CHECK IF SEEK TO PREIOUS CYLINDER
011534  001310                          BNE     MAIN            ;YES, GET NEXT AVAILIABLE DRIVE
011536  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
011544  005737  001320                  TST     BWAITQ          ;CHECK IF BUFFER WAIT QUEUE EMPTY
011550  001013                          BNE     25$             ;NO, ENQUEUE PARAMETER BLOCK
011552  004037  022440                  JSR     R0,BUFFAL       ;ALLOCATE BUFFER
011556  011600                          25$                     ;NOT ENOUGH ROOM RETURN
011560  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
011564  004737  023056                  JSR     PC,LDDATA       ;LOAD BUFFER
011570  004037  051406                  JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
011574  001324                          CINITQ                  ;  INITIATION QUEUE
011576  000667                          BR      MAIN            ;GET NEXT AVAILABLE DRIVE

011600  004037  051406          25$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON BUFFER
011604  001320                          BWAITQ                  ;  WAIT QUEUE
011606  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
011612  000661                          BR      MAIN            ;GET NEXT AVAILABLE DRIVE
```

```
                                .SBTTL  OPERATOR COMMAND DECODER

                                ;;*****************************************************************
                                ;*
                                ;*      THE LEGAL COMMANDS ARE:
                                ;*              TN-     INITIATE TESTING ON DRIVE N
                                ;*              PN-     CHANGE PARAMETER AND INITIATE TEST ON
                                ;*                      DRIVE N
                                ;*              DN-     DROP DRIVE FROM TESTING SEQUENCE
                                ;*              SN-     DEMAND PERFORMANCE SUMMARY ON DRIVE N
                                ;*              WN-     WRITE RANDOM DATA PATTERNS ON DRIVE N
                                ;*                      AND INITIATE TESTING.
                                ;*
                                ;*CALL  JSR     PC,OPRCMD
                                ;*      RETURN
                                ;*
                                ;*
                                ;*      COMMAND         ROUTINE
                                ;*      -------         -------
                                ;*
                                ;*      TN              TESTDV
                                ;*      PN              PARMDV
                                ;*      DN              DROPDV
                                ;*      SN              STATDV
                                ;*      WN              WRTEPK
                                ;*
                                ;;*****************************************************************

011614  010446                  OPRCMD: MOV     R4,-(SP)        ;STORE R4 ON STACK
011616  104401  056230                  TYPE    ,OPR007         ;TYPE "PLEASE ENTER COMMAND"
011622  104403                          RDLIN                   ;READ LINE FROM TTY
011624  012604                          MOV     (SP)+,R4        ;STORE ADDRESS OF INPUT STRING
011626  116437  000001  056065          MOVB    1(R4),OPR001    ;MOVE SECOND CHARACTER TO DRIVE
                                                                ;  REQUEST STORAGE
011634  122737  000101  056065          CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES
011642  001421                          BEQ     4$              ;YES, PROCESS ALL DRIVES
011644  122737  000060  056065          CMPB    #'0,OPR001      ;CHECK IF LEGAL DRIVE NUMBER
011652  101004                          BHI     2$              ;NO, PRINT ERROR
011654  122737  000067  056065          CMPB    #'7,OPR001      ;CHECK IF 0-7
011662  103006                          BHIS    3$              ;YES, PROCESS DRIVE COMMAND
011664  104401  056050          2$:     TYPE    ,OPR000         ;TYPE "DRIVE NUMBER N ILLEGAL?"
011670  104401  056070                  TYPE    ,OPR002
011674  012604                          MOV     (SP)+,R4        ;RESTORE R4
011676  000207                          RTS     PC              ;RETURN

011700  142737  000370  056065  3$:     BICB    #370,OPR001     ;CLEAR UNIMPORTANT BITS
011706  010546                  4$:     MOV     R5,-(SP)        ;STORE R5 ON STACK
011710  122714  000124                  CMPB    #'T,(R4)        ;CHECK IF TEST DRIVE
011714  001427                          BEQ     TESTDV          ;YES, GO TO TEST DRIVE
011716  122714  000120                  CMPB    #'P,(R4)        ;CHECK IF TEST AND CHANGE PARAMETERS
011722  001512                          BEQ     PARMDV          ;YES, GO TEST AND CHANGE PARAMETERS
011724  122714  000104                  CMPB    #'D,(R4)        ;CHECK IF DROP DRIVE
011730  001553                          BEQ     DROPDV          ;YES, GO DROP DRIVE
011732  122714  000123                  CMPB    #'S,(R4)        ;CHECK IF DEMAND STATISTICS
011736  001002                          BNE     5$
011740  000137  012440                  JMP     STATDV          ;YES, GO TO DEMAND STATISTICS
```

```
3900
3901  011744  122714  000127      5$:   CMPB   #'W,(R4)        ;CHECK IF WRITE PACK AND TEST
3902  011750  001002                    BNE    6$
3903  011752  000137  012614            JMP    WRTEPK          ;YES, GO WRITE PACK AND TEST
3904
3905  011756  111437  056042      6$:   MOVB   (R4),ILLCMD     ;STORE COMMAND FOR PRINT OUT
3906  011762  104401  056011            TYPE   ,ILLCOM         ;TYPE ERROR MESSAGE
3907  011766  012605                    MOV    (SP)+,R5        ;RESTORE R5
3908  011770  012604                    MOV    (SP)+,R4        ;RESTORE R4
3909  011772  000207                    RTS    PC              ;RETURN
```

```
                                        .SBTTL  TEST DRIVE COMMAND
                                
011774  122737  000101  056065  TESTDV: CMPB    #'A,OPROO1          ;CHECK IF ALL DRIVES ARE TO BE TESTED
012002  001027                  BNE     10$                 ;NO, DO SPECIFIC DRIVE
012004  105037  001632          CLRB    DRVCNT              ;CLEAR DRIVE COUNT
012010  005004                  CLR     R4                  ;CLEAR DRIVE COUNT
012012  016405  001524          1$:     MOV     PBLKT(R4),R5        ;GET PARAMETER BLOCK ADDRESS
012016  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
012024  001004                  BNE     2$                  ;YES, GO TO NEXT DRIVE
012026  004737  013360          JSR     PC,TEST             ;GO TEST DRIVE
012032  105237  001632          INCB    DRVCNT              ;INCREMENT DRIVE COUNT
012036  005724                  2$:     TST     (R4)+               ;ADDRESS NEXT DRIVE (ADD 2)
012040  022704  000020          CMP     #20,R4              ;CHECK IF FINISHED
012044  001362                  BNE     1$                  ;NO, SET UP NEXT DRIVE
012046  105737  001632          TSTB    DRVCNT              ;CHECK IF ANY DRIVE IN ASSIGNMENT SEQUENCE
012052  001033                  BNE     15$                 ;YES, RETURN
012054  104401  056407          TYPE    ,OPRO13             ;TYPE "ALL DRIVES CURRENTLY UNDER TEST?"
012060  000430                  BR      15$                 ;RETURN
                                
012062  113704  056065          10$:    MOVB    OPROO1,R4           ;STORE DRIVE NUMBER
012066  006304                  ASL     R4                  ;MULTIPLY DRIVE NUMBER BY 2
012070  016405  001524          MOV     PBLKT(R4),R5        ;GET PARAMETER ADDRESS
012074  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
012102  001003                  BNE     11$                 ;YES, PRINT DRIVE ALREADY ASSIGNED
012104  004737  013360          JSR     PC,TEST             ;GO TEST DRIVE
012110  000414                  BR      15$                 ;RETURN
                                
012112  013737  001452  177776  11$:    MOV     RKPRI,PS            ;LOCK OUT RK06 INTERRUPTS
012120  152737  000060  056065          BISB    #60,OPROO1          ;MAKE NUMBER ASCIZ
012126  104401  056050          TYPE    ,OPROOO             ;TYPE "DRIVE N ALREADY ASSIGNED?"
012132  104401  056116          TYPE    ,OPRO04
012136  005037  177776          CLR     PS                  ;ALLOW RK06 INTERRUPTS
012142  012605                  15$:    MOV     (SP)+,R5            ;RESTORE R5
012144  012604                  MOV     (SP)+,R4            ;RESTORE R4
012146  000207                  RTS     PC                  ;RETURN
                                
                                        .SBTTL  CHANGE PARAMETERS COMMAND
                                
012150  122737  000101  056065  PARMDV: CMPB    #'A,OPROO1          ;CHECK IF PARAMETERS FOR ALL DRIVES
012156  001020                  BNE     10$                 ;NO, DO SPECIFIC DRIVE
012160  005004                  CLR     R4                  ;CLEAR DRIVE COUNT
012162  016405  001524          1$:     MOV     PBLKT(R4),R5        ;GET PARAMETER BLOCK ADDRESS
012166  004737  016170          JSR     PC,PARM             ;ASK FOR PARAMETERS
012172  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
012200  001002                  BNE     2$                  ;YES, GO TO NEXT DRIVE
012202  004737  013360          JSR     PC,TEST             ;GO TEST DRIVE
012206  005724                  2$:     TST     (R4)+               ;ADDRESS NEXT DRIVE (ADD 2)
012210  022704  000020          CMP     #20,R4              ;CHECK IF FINISHED
012214  001362                  BNE     1$                  ;NO, SET UP NEXT DRIVE
012216  000415                  BR      15$                 ;RETURN
                                
012220  113704  056065          10$:    MOVB    OPROO1,R4           ;STORE DRIVE NUMBER
012224  006304                  ASL     R4                  ;MULTIPLY DRIVE NUMBER BY 2
012226  016405  001524          MOV     PBLKT(R4),R5        ;GET PARAMETER ADDRESS
012232  004737  016170          JSR     PC,PARM             ;GET NEW PARAMETERS
012236  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
```

```
3966  012244  001002                        BNE     15$                     ;YES, RETURN
3967  012246  004737  013360                JSR     PC,TEST                 ;GO TEST DRIVE
3968  012252  012605              15$:      MOV     (SP)+,R5                ;RESTORE R5
3969  012254  012604                        MOV     (SP)+,R4                ;RESTORE R4
3970  012256  000207                        RTS     PC                      ;RETURN
3971
3972                                  .SBTTL  DROP DRIVE COMMAND
3973
3974  012260  122737  000101  056065  DROPDV: CMPB    #'A,OPR001              ;CHECK IF ALL DRIVES ARE TO BE DROPPED
3975  012266  001030                        BNE     10$                     ;NO, DO SPECIFIC DRIVE
3976  012270  105037  001632                CLRB    DRVCNT                  ;CLEAR DRIVE COUNT
3977  012274  005004                        CLR     R4                      ;CLEAR DRIVE COUNT
3978  012276  016405  001524        1$:     MOV     PBLKT(R4),R5            ;GET PARAMETER BLOCK ADDRESS
3979  012302  032765  000001  000014        BIT     #DRVUSE,P.PRST(R5)      ;CHECK IF DRIVE IN USE
3980  012310  001405                        BEQ     2$                      ;NO, GO TO NEXT DRIVE
3981  012312  052765  002000  000014        BIS     #DRPDRV,P.PRST(R5)      ;INDICATE DRIVE IS TO BE DROPPED
3982  012320  105237  001632                INCB    DRVCNT                  ;INCREMENT DRIVE COUNT
3983  012324  005724              2$:       TST     (R4)+                   ;ADDRESS NEXT DRIVE (ADD 2)
3984  012326  022704  000020                CMP     #20,R4                  ;CHECK IF FINISHED
3985  012332  001361                        BNE     1$                      ;NO, SET UP NEXT DRIVE
3986  012334  105737  001632                TSTB    DRVCNT                  ;CHECK IF ANY DRIVES AVAILIABLE
3987  012340  001034                        BNE     15$                     ;YES, RETURN
3988  012342  104401  056451                TYPE    ,OPR014                 ;TYPE "NO DRIVES IN USE?"
3989  012346  000431                        BR      15$                     ;RETURN
3990
3991  012350  113704  056065        10$:    MOVB    OPR001,R4               ;STORE DRIVE NUMBER
3992  012354  006304                        ASL     R4                      ;MULTIPLY DRIVE NUMBER BY 2
3993  012356  016405  001524                MOV     PBLKT(R4),R5            ;GET PARAMETER ADDRESS
3994  012362  032765  000001  000014        BIT     #DRVUSE,P.PRST(R5)      ;CHECK IF DRIVE ASSIGNED
3995  012370  001404                        BEQ     11$                     ;NO, PRINT DRIVE N NOT ASSIGNED
3996  012372  052765  002000  000014        BIS     #DRPDRV,P.PRST(R5)      ;INDICATE DRIVE IS TO BE DROPPED
3997  012400  000414                        BR      15$                     ;RETURN
3998
3999  012402  013737  001452  177776  11$:    MOV     RKPRI,PS                ;LOCK OUT RK06 INTERRUPTS
4000  012410  152737  000060  056065        BISB    #60,OPR001              ;MAKE NUMBER ASCII
4001  012416  104401  056050                TYPE    ,OPR000                 ;TYPE "DRIVE N NOT ASSIGNED?"
4002  012422  104401  056142                TYPE    ,OPR005
4003  012426  005037  177776                CLR     PS                      ;ALLOW RK06 INTERRUPTS
4004  012432  012605              15$:      MOV     (SP)+,R5                ;RESTORE R5
4005  012434  012604                        MOV     (SP)+,R4                ;RESTORE R4
4006  012436  000207                        RTS     PC                      ;RETURN
4007
4008                                  .SBTTL  GET DRIVE STATISTICS COMMAND
4009
4010  012440  122737  000101  056065  STATDV: CMPB    #'A,OPR001              ;CHECK IF STATISTICS FOR ALL DRIVES
4011  012446  001027                        BNE     10$                     ;NO, DP SPECIFIC DRIVE
4012  012450  105037  001632                CLRB    DRVCNT                  ;CLEAR DRIVE COUNT
4013  012454  005004                        CLR     R4                      ;CLEAR DRIVE COUNT
4014  012456  016405  001524        1$:     MOV     PBLKT(R4),R5            ;GET PARAMETER BLOCK ADDRESS
4015  012462  032765  000001  000014        BIT     #DRVUSE,P.PRST(R5)      ;CHECK IF DRIVE IN USE
4016  012470  001404                        BEQ     2$                      ;NO, GO TO NEXT DRIVE
4017  012472  004737  013102                JSR     PC,STAT                 ;GATHER STATISTICS
4018  012476  105237  001632                INCB    DRVCNT                  ;INCREMENT DRIVE COUNT
4019  012502  005724              2$:       TST     (R4)+                   ;ADDRESS NEXT DRIVE (ADD 2)
4020  012504  022704  000020                CMP     #20,R4                  ;CHECK IF FINISHED
4021  012510  001362                        BNE     1$                      ;NO, SET UP NEXT DRIVE
```

```
4022  012512  105737  001632                  TSTB    DRVCNT          ;CHECK IF ANY DRIVES AVAILIABLE
4023  012516  001033                          BNE     15$             ;YES, RETURN
4024  012520  104401  056451                  TYPE    ,OPR014         ;TYPE "NO DRIVES IN USE?"
4025  012524  000430                          BR      15$             ;RETURN
4026
4027  012526  113704  056065          10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
4028  012532  006304                          ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
4029  012534  016405  001524                  MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
4030  012540  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
4031  012546  001403                          BEQ     11$             ;PRINT DRIVE N NOT ASSIGNED
4032  012550  004737  013102                  JSR     PC,STAT         ;GO GATHER DRIVE STATISTICS
4033  012554  000414                          BR      15$             ;RETURN
4034
4035  012556  013737  001452  177776  11$:    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
4036  012564  152737  000060  056065          BISB    #60,OPR001      ;MAKE NUMBER ASCII
4037  012572  104401  056050                  TYPE    ,OPR000         ;TYPE "DRIVE N NOT ASSIGNED?"
4038  012576  104401  056142                  TYPE    ,OPR005
4039  012602  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4040  012606  012605                  15$:    MOV     (SP)+,R5        ;RESTORE R5
4041  012610  012604                          MOV     (SP)+,R4        ;RESTORE R4
4042  012612  000207                          RTS     PC              ;RETURN
4043
4044                                  .SBTTL  WRITE PACK AND TEST COMMAND
4045
4046  012614  122737  000101  056065  WRTEPK: CMPB    #'A,OPR001      ;CHECK IF ALL DRIVES ARE TO BE TESTED
4047  012622  001032                          BNE     10$             ;NO, DO SPECIFIC DRIVE
4048  012624  105037  001632                  CLRB    DRVCNT          ;CLEAR DRIVE COUNT
4049  012630  005004                          CLR     R4              ;CLEAR DRIVE COUNT
4050  012632  016405  001524          1$:     MOV     PBLKT(R4),R5    ;GET PARAMETER BLOCK ADDRESS
4051  012636  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE IN USE
4052  012644  001007                          BNE     2$              ;YES, GO TO NEXT DRIVE
4053  012646  112765  000020  000210          MOVB    #BIT4,P.ASSN(R5) ;SET FLAG FOR WRITE PACK
4054  012654  004737  013360                  JSR     PC,TEST         ;GO ASSIGN DRIVE
4055  012660  105237  001632                  INCB    DRVCNT          ;INCREMENT DRIVE COUNT
4056  012664  005724                  2$:     TST     (R4)+           ;ADDRESS NEXT DRIVE (ADD 2)
4057  012666  022704  000020                  CMP     #20,R4          ;CHECK IF FINISHED
4058  012672  001357                          BNE     1$              ;NO, SET UP NEXT COMMAND
4059  012674  105737  001632                  TSTB    DRVCNT          ;CHECK IF ANY DRIVES NOT IN USE
4060  012700  001436                          BEQ     15$             ;NO, RETURN
4061  012702  104401  056407                  TYPE    ,OPR013         ;TYPE "ALL DRIVES CURRENTLY UNDER TEST"
4062  012706  000433                          BR      15$             ;RETURN
4063
4064  012710  113704  056065          10$:    MOVB    OPR001,R4       ;STORE DRIVE NUMBER
4065  012714  006304                          ASL     R4              ;MULTIPLY DRIVE NUMBER BY 2
4066  012716  016405  001524                  MOV     PBLKT(R4),R5    ;GET PARAMETER ADDRESS
4067  012722  032765  000001  000014          BIT     #DRVUSE,P.PRST(R5)  ;CHECK IF DRIVE ASSIGNED
4068  012730  001006                          BNE     11$             ;YES, PRINT DRIVE ALREADY ASSIGNED
4069  012732  112765  000020  000210          MOVB    #BIT4,P.ASSN(R5) ;SET FLAG FOR WRITE PACK
4070  012740  004737  013360                  JSR     PC,TEST         ;GO ASSIGN DRIVE
4071  012744  000414                          BR      15$             ;RETURN
4072
4073  012746  013737  001452  177776  11$:    MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
4074  012754  152737  000060  056065          BISB    #60,OPR001      ;MAKE NUMBER ASCII
4075  012762  104401  056050                  TYPE    ,OPR000         ;TYPE "DRIVE N ALREADY ASSIGNED?"
4076  012766  104401  056116                  TYPE    ,OPR004
4077  012772  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
```

```
 4078  012776  012605                15$:    MOV     (SP)+,R5        ;RESTORE R5
 4079  013000  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4080  013002  000207                        RTS     PC              ;RETURN
 4081
```

```
                                .SBTTL  DROP DRIVE AND GATHER STATISTICS ROUTINES
                        013004  DROP:   MOV     PS,-(SP)        ;STORE PSW
177776  013010  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT TTY AND RK06 INTERRUPTS
        013016  005065  000014          CLR     P.PRST(R5)      ;CLEAR PROGRAM STAUS REGISTER
        013022  005065  000212          CLR     P.ERR(R5)       ;CLEAR ERROR FLAGS
056357  013026  116537  000000  056357  MOVB    P.DRVN(R5),OPRO11 ;LOAD DRIVE NUMBER
056357  013034  152737  000060  056357  BISB    #60,OPRO11      ;MAKE IT ASCII
        013042  104401  056340          TYPE    ,OPRO10         ;TYPE "DRIVE N HAS BEEN DROPPED
        013046  104401  056162          TYPE    ,OPRO06         ;  FROM TEST SEQUENCE"
000210  013052  132765  000001  000210  BITB    #BIT0,P.ASSN(R5) ;SEE IF SIZING FOR DRIVE
        013060  001404                  BEQ     2$              ;BR IF NOT
        013062  105065  000210          CLRB    P.ASSN(R5)      ;CLEAR ASSINGMENT CODE BITS
        013066  000137  013352          JMP     STAT01
        013072  105065  000210  2$:     CLRB    P.ASSN(R5)      ;CLEAR ASSINGMENT CODE BITS
        013076  000137  013114          JMP     STAT00

        013102  013746  177776  STAT:   MOV     PS,-(SP)        ;STORE PSW
177776  013106  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT TTY AND RK06 INTERRUPTS
        013114  104401  056526  STAT00: TYPE    ,DRVSTT         ;**DRIVE STATISTICS
056357  013120  116537  000000  056357  MOVB    P.DRVN(R5),OPRO11 ;LOAD DRIVE NUMBER FOR PRINT OUT
056357  013126  152737  000060  056357  BISB    #60,OPRO11      ;MAKE IT ASCII
        013134  104401  056340          TYPE    ,OPRO10         ;TYPE "DRIVE N"
        013140  004737  027616          JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
        013144  004737  043670          JSR     PC,PRITIM       ;PRINT TIME IF CLOCK AVAILIABLE
        013150  104401  056604          TYPE    ,STTO01         ;TYPE "ORDERS PERFORMED"
        013154  010546                  MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
000152  013156  062716  000152          ADD     #P.NODR,(SP)    ;CALCULATE ADDRESS FOR PRINT OUT
        013162  004737  055174          JSR     PC,$DB2D        ;CONVERT TO ASCII
        013166  104401  056626          TYPE    ,STTO02         ;TYPE "WORDS WRITTEN*65K"
        013172  010546                  MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
000160  013174  062716  000160          ADD     #P.NWRT+2,(SP)  ;CALCULATE ADDRESS FOR PRINT OUT
        013200  004737  055174          JSR     PC,$DB2D        ;CONVERT TO ASCII
        013204  104401  056653          TYPE    ,STTO03         ;TYPE "WORDS READ*65K"
        013210  010546                  MOV     R5,-(SP)        ;GET PARAMETER BLOCK ADDRESS
000166  013212  062716  000166          ADD     #P.NRD+2,(SP)   ;CALCULATE ADDRESS FOR PRINT OUT
        013216  004737  055174          JSR     PC,$DB2D        ;CONVERT TO ASCII
        013222  104401  056675          TYPE    ,STTO04         ;TYPE "SOFT DATA ERRORS"
                                                                ;  "ECC"
000176  013226  016546  000176          MOV     P.SECC(R5),-(SP) ;STORE SOFT ECC ERROR COUNT
        013232  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
        013236  104401  056730          TYPE    ,STTO05         ;TYPE "REREAD"
000172  013242  016546  000172          MOV     P.SRRD(R5),-(SP) ;SAVE SOFT REREAD CORRECTABLE
        013246  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
        013252  104401  056744          TYPE    ,STTO06         ;TYPE "OFFSET"
000174  013256  016546  000174          MOV     P.SOFF(R5),-(SP) ;SAVE SOFT OFFSET CORRECTABLE
        013262  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
        013266  104401  056760          TYPE    ,STTO07         ;TYPE "HARD DATA ERRORS"
000200  013272  016546  000200          MOV     P.HARD(R5),-(SP) ;SAVE HARD DATA ERRORS
        013276  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
        013302  104401  057004          TYPE    ,STTO08         ;TYPE "SEEK INCOMPLETES
000202  013306  016546  000202          MOV     P.NSKI(R5),-(SP) ;TYPE NUMBER OF SEEK INCOMPLETES
        013312  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
        013316  104401  057030          TYPE    ,STTO09         ;TYPE "OPERATION INCOMPLETES"
000204  013322  016546  000204          MOV     P.NOPI(R5),-(SP) ;SAVE NUMBER OF OPERATION INCOMPLETES
        013326  004737  055132          JSR     PC,$SB2D        ;CONVERT TO DECIMAL
```

```
 4138  013332  104401  057076                 TYPE    ,STT010                 ;TYPE "OTHER ERRORS"
 4139  013336  016546  000206                 MOV     P.NER(R5),-(SP)         ;SAVE NUMBER OF OTHER ERRORS
 4140  013342  004737  055132                 JSR     PC,$S82D                ;CONVERT TO DECIMAL
 4141  013346  104401  001165                 TYPE    ,$CRLF                  ;TYPE <CR><LF>
 4142  013352  012637  177776       STAT01:   MOV     (SP)+,PS                ;RESTORE PSW
 4143  013356  000207                 RTS     PC                              ;RETURN
 4144
```

```
                                        .SBTTL  INITIATE DRIVE ASSIGNMENT SEQUENCE

013360  116546  000210        TEST:   MOVB    P.ASSN(R5),-(SP) ;SAVE ASSIGNMENT CODE
013364  010446                        MOV     R4,-(SP)        ;STORE R4 ON STACK
013366  010346                        MOV     R3,-(SP)        ;STORE R3 ON STACK
013370  010503                        MOV     R5,R3           ;LOAD PARAMETER BLOCK ADDRESS INTO R3
013372  062703  000122                ADD     #P.RDPT,R3      ;CALCULATE BEGINNING OF STATISTICS AND
                                                                ;  ERROR AREA
013376  010504                        MOV     R5,R4           ;LOAD PARAMETER BLOCK ADDRESS INTO R4
013400  062704  000232                ADD     #P.EXAR,R4      ;CALCULATE END OF STATISTICS AND
                                                                ;  ERROR AREA
013404  005023                1$:     CLR     (R3)+           ;CLEAR STATISTIC AND ERROR AREA
013406  020304                        CMP     R3,R4           ;CHECK IF FINISHED
013410  001375                        BNE     1$              ;NO, CONTINUE
013412  116504  000000                MOVB    P.DRVN(R5),R4   ;STORE DRIVE NUMBER
013416  156437  001514  001513        BISB    INTMSK(R4),O.OVER ;SET UP FOR OVERLAPPING SEEKS
013424  012603                        MOV     (SP)+,R3        ;RESTORE R3
013426  012604                        MOV     (SP)+,R4        ;RESTORE R4
013430  112665  000210                MOVB    (SP)+,P.ASSN(R5) ;RESTORE ASSIGNMENT CODE

013434  105065  000117        TEST1:  CLRB    P.ECMP(R5)      ;CLEAR ECC COMPARE FLAG
013440  005037  001500                CLR     OPTFLG          ;CLEAR CMND OPTIMIZER FLAG
013444  152765  000001  000210        BISB    #BIT0,P.ASSN(R5) ;SET READ STATUS COMMAND FLAG
013452  052765  000001  000014        BIS     #DRVUSE,P.PRST(R5) ;SET DRIVE IN USE
013460  112765  000141  000001        MOVB    #RDSTAT,P.CMND(R5) ;READ ALL DRIVE STATUS
013466  112765  000141  000123        MOVB    #RDSTAT,P.RCMD(R5)
013474  004037  051406                JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
013500  001324                        CINITQ                  ;  COMMAND INITIATION QUEUE
013502  000207                        RTS     PC              ;RETURN
```

```
                                        .SBTTL  DRIVE ASSIGNMENT SEQUENCE

013504  132765  000001  000210  ASNORM: BITB    #BIT0,P.ASSN(R5) ;CHECK IF READ STATUS
013512  001467                  BEQ     ASN4$           ;NO, CHECK IF START SPINDLE
013514  005737  027156          TST     NEWFLG          ;SEE IF DTYE ERROR ON ASSIGNING
013520  001404                  BEQ     1$              ;BR IF NO
013522  005037  027156          CLR     NEWFLG          ;ELSE CLEAR DTYE ON ASSIGN FLAG
013526  000137  013434          JMP     TESTI           ;& REPEAT SELECT FOR DRIVE THAT HAD DTYE
013532  142765  000001  000210  1$:     BICB    #BIT0,P.ASSN(R5) ;CLEAR READ STATUS COMMAND ISSUED
013540  016565  000054  000224          MOV     P.A11(R5),P.SERL(R5) ;STORE SERIAL NUMBER
013546  042765  100007  000224          BIC     #^C<M.SER>,P.SERL(R5) ;KEEP SERIAL NUMBER
013554  006265  000224                  ASR     P.SERL(R5)      ;SHIFT 3 BITS RIGHT
013560  006265  000224                  ASR     P.SERL(R5)
013564  006265  000224                  ASR     P.SERL(R5)
013570  032765  000200  000036          BIT     #DRDY,P.DS(R5)  ;CHECK IF DRIVE READY
013576  001415                  BEQ     ASN2$           ;NO, ISSUE START SPINDLE
013600  116565  000123  000135  ASN1$:  MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
013606  112765  000103  000001          MOVB    #PACK,P.CMND(R5) ;ISSUE PACK ACKNOWLEDGE
013614  112765  000103  000123          MOVB    #PACK,P.RCMD(R5)
013622  152765  000004  000210          BISB    #BIT2,P.ASSN(R5) ;SET PACK ACKNOWLEDGE FLAG
013630  000414                  BR      ASN3$           ;ISSUE PACK ACKNOWLEDGE AND WAIT

013632  116565  000123  000135  ASN2$:  MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
013640  112765  000111  000001          MOVB    #SRTSPL,P.CMND(R5) ;ISSUE START SPINDLE
013646  112765  000111  000123          MOVB    #SRTSPL,P.RCMD(R5)
013654  152765  000002  000210          BISB    #BIT1,P.ASSN(R5) ;SET START SPINDLE FLAG
013662  004037  051406          ASN3$:  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
013666  001324                  CINITQ                  ;  COMMAND INITIATION QUEUE
013670  000207                  RTS     PC              ;RETURN

013672  132765  000002  000210  ASN4$:  BITB    #BIT1,P.ASSN(R5) ;CHECK IF START SPINDLE
013700  001414                  BEQ     15$             ;NO, CHECK IF PACK ACKNOWLEDGE
013702  142765  000002  000210          BICB    #BIT1,P.ASSN(R5) ;CLEAR START SPINDLE FLAG
013710  032765  000200  000036          BIT     #DRDY,P.DS(R5)  ;CHECK IF DRIVE READY
013716  001330                  BNE     ASN1$           ;YES, ISSUE PACK ACKNOWLEDGE
013720  052765  002000  000212          BIS     #BIT10,P.ERR(R5) ;SET DRIVE NOT READY AFTER START SPINDLE
013726  000137  025312          JMP     ABNORM          ;REPORT ERROR

013732  132765  000004  000210  15$:    BITB    #BIT2,P.ASSN(R5) ;CHECK IF PACK ACKNOWLEDGE
013740  001431                  BEQ     18$             ;NO, CHECK IF RECAL
013742  142765  000004  000210          BICB    #BIT2,P.ASSN(R5) ;CLEAR PACK ACKNOWLEDGE FLAG
013750  032765  000100  000036          BIT     #VV,P.DS(R5)    ;CHECK IF VOLUME VALID IS SET
013756  001005                  BNE     16$             ;YES, CHECK IF PACK IS TO BE WRITTEN
013760  052765  004000  000212          BIS     #BIT11,P.ERR(R5) ;SET VOLUME VALID NOT SET AFTER PACK
                                                        ;  ACKNOWLEDGE
013766  000137  025312          JMP     ABNORM          ;REPORT ERROR

013772  116565  000123  000135  16$:    MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
014000  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;ISSUE RECAL
014006  112765  000113  000123          MOVB    #RECAL,P.RCMD(R5)
014014  152765  000100  000210          BISB    #BIT6,P.ASSN(R5) ;SET RECAL ISSUED
014022  000717                  BR      ASN3$

014024  132765  000100  000210  18$:    BITB    #BIT6,P.ASSN(R5) ;CHECK IF RECAL
014032  001457                  BEQ     23$             ;NO, CHECK IF READ PACK SERIAL NUMBER
014034  142765  000100  000210          BICB    #BIT6,P.ASSN(R5) ;RESET RECAL ISSUED
```

```
4230  014042  152765  000010  000210          BISB    #BIT3,P.ASSN(R5) ;SET ASSIGNMENT CODE FOR PACK
4231                                                                    ;  SERIAL NUMBER READ
4232  014050  012765  001000  000004          MOV     #1000,P.SECT(R5) ;  TRACK 2 SECTOR 0
4233  014056  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
4234  014064  112765  000121  000001  20$:    MOVB    #RDDATA,P.CMND(R5) ;LOAD READ COMMAND FOR PACK
4235  014072  112765  000121  000123          MOVB    #RDDATA,P.RCMD(R5) ;  SERIAL NUMBER
4236  014100  012765  177774  000012          MOV     #-4,P.WC(R5)    ;LOAD WORD COUNT FOR THE FIRST FOUR WORDS
4237  014106  004737  027074                  JSR     PC,GETNUM       ;GET CORRECT END CYL NUM
4238  014112  013765  027154  000002          MOV     LSTCYL,P.CYLN(R5) ;SET CYLINDER TO 632/1456
4239  014120  013765  027154  000124          MOV     LSTCYL,P.RCYL(R5)
4240  014126  016565  000004  000126          MOV     P.SECT(R5),P.RSEC(R5) ;STORE TRACK AND SECTOR
4241  014134  005737  001320                  TST     BWAITQ          ;CHECK IF BUFFER WAIT QUEUE IS EMPTY
4242  014140  001010                          BNE     22$             ;NO, ENQUEUE PARAMETER BLOCK IN BUFFER
4243                                                                  ;  WAIT QUEUE
4244  014142  004037  022440                  JSR     R0,BUFFAL       ;ALLOCATE BUFFER
4245  014146  014162                          22$                     ;NOT ENOUGH ROOM RETURN
4246  014150  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4247  014154  004737  023056                  JSR     PC,LDDATA       ;CLEAR BUFFER
4248  014160  000640                          BR      ASN3$           ;ISSUE COMMAND
4249
4250  014162  004037  051406          22$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON BUFFER
4251  014166  001320                          BWAITQ                  ;  WAIT QUEUE
4252  014170  000207                          RTS     PC              ;RETURN
4253
4254  014172  032777  040000  164740  23$:    BIT     #SW14,@SWR      ;CHECK IS CYCLE ON OPERATION
4255  014200  001542                          BEQ     43$             ;NO, CONTINUE WITH ASSIGNMENT PROCESS
4256  014202  105765  000211                  TSTB    P.SEEK(R5)      ;CHECK IF SEEK TO PREVIOUS POSITION
4257  014206  001444                          BEQ     28$             ;NO, SERVICE INTERRUPT
4258  014210  105065  000211                  CLRB    P.SEEK(R5)      ;CLEAR FLAG
4259  014214  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST COMMAND
4260  014222  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5) ;LOAD CYLINDER ADDRESS
4261  014230  016565  000126  000004          MOV     P.RSEC(R5),P.SECT(R5) ;LOAD PREVIOUS TRACK AND SECTOR
4262  014236  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5) ;LOAD PREVIOUS WORD COUNT
4263  014244  116565  000122  000121          MOVB    P.RDPT(R5),P.DPAT(R5) ;LOAD PREVIOUS PATTERN
4264  014252  132765  000010  000210          BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
4265  014260  001003                          BNE     25$             ;YES, DO NOT SET WRITE BEFORE WRITE CHECK
4266  014262  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;SET WRITE BEFORE WRITE CHECK
4267  014270  005737  001320          25$:    TST     BWAITQ          ;CHECK IF ANY DRIVE WAITING FOR BUFFER
4268  014274  001332                          BNE     22$             ;YES, ENQUEUE PARAMETER BLOCK
4269  014276  004037  022440                  JSR     R0,BUFFAL       ;GO ALLOCATE BUFFER
4270  014302  014162                          22$                     ;NOT ENOUGH ROOM RETURN
4271  014304  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4272  014310  004737  023056                  JSR     PC,LDDATA       ;GO LOAD DATA
4273  014314  000137  013662                  JMP     ASN3$           ;GO ISSUE COMMAND
4274
4275  014320  004037  025032          28$:    JSR     R0,CHKADD       ;CHECK SECTOR, TRACK, CYLINDER,
4276                                                                  ;  WORD COUNT, AND BUS ADDRESS
4277  014324  015232                          60$                     ;ERROR RETURN
4278  014326  132765  000010  000210          BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
4279  014334  001424                          BEQ     35$             ;NO, RELEASE BUFFER
4280  014336  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
4281  014340  016504  000010                  MOV     P.BALO(R5),R4   ;LOAD R4 TO GET PACK SERIAL NUMBER
4282  014344  012465  000226                  MOV     (R4)+,P.PKSR(R5) ;GET PACK SERIAL NUMBER
4283  014350  012465  000230                  MOV     (R4)+,P.PKSR+2(R5)
4284  014354  005724                          TST     (R4)+           ;ADJUST R4 TO ADDRESS ALLIGNMENT INDICATION
4285  014356  005714                          TST     (R4)            ;CHECK IF DATA PACK
```

```
4286  014360  001423                        BEQ     37$                     ;YES, CONTINUE
4287  014362  012604                30$:    MOV     (SP)+,R4                ;RESTORE R4
4288  014364  004737  022642                JSR     PC,BUFREL               ;RELEASE BUFFER
4289  014370  104401  056473                TYPE    ,OPR015                 ;TYPE "ALLIGNMENT PACT IN DRIVE"
4290  014374  004737  013004                JSR     PC,DROP                 ;GO DROP DRIVE
4291  014400  004737  022342                JSR     PC,GETBUF               ;GO GET BUFFER FOR NEXT COMMAND
4292  014404  000207                        RTS     PC                      ;RETURN
4293
4294  014406  032765  000200  000014  35$:  BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK HAS BEEN ISSUED
4295  014414  001406                        BEQ     38$                     ;YES, GO RELEASE BUFFER
4296  014416  042765  000200  000014        BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE BEFORE WRITE CHECK
4297  014424  000137  013662                JMP     ASN3$                   ;GO ISSUE COMMAND
4298
4299  014430  012604                37$:    MOV     (SP)+,R4                ;RESTORE R4
4300  014432  004737  022642        38$:    JSR     PC,BUFREL               ;RELEASE BUFFER
4301  014436  032765  002000  000014        BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
4302  014444  001402                        BEQ     40$                     ;NO, SEEK TO PREVIOUS POSITION
4303  014446  000137  016040                JMP     WV1$                    ;GO DROP DRIVE
4304
4305  014452  112765  177777  000211  40$:  MOVB    #-1,P.SEEK(R5)  ;SET FLAG
4306  014460  016565  000136  000002        MOV     P.LCYL(R5),P.CYLN(R5) ;LOAD LAST CYLINDER
4307  014466  016565  000140  000004        MOV     P.LSEC(R5),P.SECT(R5) ;LOAD LAST SECTOR AND TRACK
4308  014474  112765  000117  000001        MOVB    #SEEK,P.CMND(R5) ;ISSUE SEEK
4309  014502  000137  013662                JMP     ASN3$                   ;GO SEEK TO PREVIOUS POSITION
4310
4311  014506  132765  000010  000210  43$:  BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
4312  014514  001002                        BNE     45$                     ;YES, SET UP FOR READ
4313  014516  000137  015234                JMP     WRTVER                  ;CHECK IF WRITE PACK
4314
4315  014522  132765  000200  00021C  45$:  BITB    #BIT7,P.ASSN(R5) ;CHECK IF RECAL AND RETRY
4316                                                                        ;  READ OF PACK SERIAL NUMBER
4317  014530  001427                        BEQ     48$                     ;NO, SERVICE READ OF PACK SERIAL NUMBER
4318  014532  142765  000200  000210        BICB    #BIT7,P.ASSN(R5) ;RESET RECAL FOR REREAD OF PACK SERIAL NO.
4319  014540  116565  000123  000135        MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE LAST COMMAND
4320  014546  016565  000124  000136        MOV     P.RCYL(R5),P.LCYL(R5) ;STORE LAST CYLINDER
4321  014554  016565  000126  000140        MOV     P.RSEC(R5),P.LSEC(R5) ;STORE LAST SECTOR AND TRACK
4322  014562  016565  000130  000142        MOV     P.RWC(R5),P.LWC(R5) ;STORE LAST WORD COUNT
4323  014570  062765  000002  000126        ADD     #2,P.RSEC(R5)   ;TRY NEXT 16-BIT FORMAT SECTOR
4324  014576  016565  000126  000004        MOV     P.RSEC(R5),P.SECT(R5) ;STORE SECOTOR TO BE READ
4325  014604  000137  014064                JMP     20$                     ;GO ISSUE READ OF PACK SERIAL NUMBER
4326
4327  014610  142765  000010  000210  48$:  BICB    #BIT3,P.ASSN(R5) ;CLEAR FLAG
4328  014616  004037  025032                JSR     R0,CHKADD               ;CHECK IF DATA TRANSFER CORRECT
4329  014622  015232                        60$                             ;ERROR RETURN
4330  014624  010446                        MOV     R4,-(SP)                ;STORE R4
4331  014626  016504  000132                MOV     P.RBAL(R5),R4   ;LOAD R4 TO ADDRESS SERIAL NUMBER
4332  014632  012465  000226                MOV     (R4)+,P.PKSR(R5) ;LOAD LEAST SIGNIFICANT BITS
4333                                                                        ;  OF SERIAL NUMBER
4334  014636  012465  000230                MOV     (R4)+,P.PKSR+2(R5) ;LOAD MOST SIGNIFICANT BITS
4335                                                                        ;  OF SERIAL NUMBER
4336  014642  005724                        TST     (R4)+                   ;ADJUST R4 TO ADDRESS ALLIGNMENT INDICATION
4337  014644  005714                        TST     (R4)                    ;CHECK IF ALLIGNMENT PACK
4338  014646  001245                        BNE     30$                     ;YES, DROP DRIVE
4339  014650  012604                        MOV     (SP)+,R4                ;RESTORE R4
4340  014652  004737  022642                JSR     PC,BUFREL               ;RELEASE BUFFER
4341  014656  132765  000020  000210        BITB    #BIT4,P.ASSN(R5) ;CHECK IF WRITE PACK
```

```
4342  014664  001042                        BNE     55$                 ;YES, SET UP PATTERN AND ALLOCATE BUFFER
4343  014666  105765  000076                TSTB    P.RATE(R5)          ;CHECK IF READ ONLY MODE
4344  014672  001413                        BEQ     52$                 ;YES, START TESTING
4345  014674  032765  004000  000036        BIT     #WRL,P.DS(R5)       ;CHECK IF WRITE LOCKED
4346  014702  001407                        BEQ     52$                 ;NO, START TESTING
4347
4348  014704  104401  056362          50$:  TYPE    ,OPR012             ;TYPE "DRIVE WRITE LOCKED"
4349  014710  004737  013004                JSR     PC,DROP             ;DROP DRIVE FROM TEST SEQUENCE
4350  014714  004737  022342                JSR     PC,GETBUF           ;GET BUFFER FOR NEXT COMMAND
4351  014720  000207                        RTS     PC                  ;RETURN
4352
4353  014722  116537  000000  056357  52$:  MOVB    P.DRVN(R5),OPR011   ;LOAD DRIVE NUMBER
4354  014730  152737  000060  056357        BISB    #60,OPR011          ;MAKE IT ASCII
4355  014736  104401  056340                TYPE    ,OPR010             ;TYPE "DRIVE NUMBER N UNDER TEST"
4356  014742  104401  056103                TYPE    ,OPR003
4357  014746  004737  027616                JSR     PC,PRISER           ;PRINT DRIVE AND PACK SERIAL NUMBERS
4358  014752  004737  043670                JSR     PC,PRITIM           ;PRINT TIME
4359  014756  104401  001165                TYPE    ,$CRLF
4360  014762  004037  051406                JSR     R0,Q.PUSH           ;PUT PARAMETER BLOCK IN DRIVE
4361  014766  001314                        AVAILQ                      ;  AVAILIABLE QUEUE
4362  014770  000207                        RTS     PC                  ;RETURN
4363
4364                                  .SBTTL  WRITE WHOLE PACK WITH RANDOM DATA
4365
4366  014772  032765  004000  000036  55$:  BIT     #WRL,P.DS(R5)       ;CHECK IF WRITE LOCKED
4367  015000  001341                        BNE     50$                 ;YES, DROP DRIVE
4368  015002  005037  177776                CLR     PS                  ;ALLOW RK06 INTERRUPTS
4369  015006  116565  000123  000135        MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE LAST COMMAND
4370  015014  016565  000124  000136        MOV     P.RCYL(R5),P.LCYL(R5) ;STORE LAST CYLINDER
4371  015022  016565  000126  000140        MOV     P.RSEC(R5),P.LSEC(R5) ;STORE LAST TRACK AND SECTOR
4372  015030  016565  000130  000142        MOV     P.RWC(R5),P.LWC(R5) ;STORE LAST WORD COUNT
4373  015036  112765  000131  000001        MOVB    #WRTCHK,P.CMND(R5) ;LOAD WRITE COMMAND
4374  015044  112765  000131  000123        MOVB    #WRTCHK,P.RCMD(R5)
4375  015052  052765  000200  000014        BIS     #W.WCK,P.PRST(R5) ;SET WRITE BEFORE WRITE CHECK
4376  015060  005065  000002                CLR     P.CYLN(R5)          ;CLEAR CYLINDER ADDRESS
4377  015064  005065  000124                CLR     P.RCYL(R5)
4378  015070  005065  000004                CLR     P.SECT(R5)          ;CLEAR TRACK AND SECTOR
4379  015074  005065  000126                CLR     P.RSEC(R5)
4380  015100  105065  000121                CLRB    P.DPAT(R5)          ;LOAD RANDOM PATTERN
4381  015104  013765  001354  000012        MOV     MAXBUF,P.WC(R5)     ;LOAD WORD COUNT
4382  015112  042765  000377  000012        BIC     #377,P.WC(R5)       ;MAKE IT AN EVEN NUMBER OF SECTORS
4383  015120  001003                        BNE     57$                 ;IF NOT ZERO USE THIS AS WORD COUNT
4384  015122  012765  000400  000012        MOV     #256.,P.WC(R5)      ;TRANSFER AT LEAST 256 WORDS
4385  015130  062765  000002  000152  57$:  ADD     #2,P.NODR(R5)       ;INCREMENT NUMBER OF ORDERS
4386  015136  066565  000012  000156        ADD     P.WC(R5),P.NWRT(R5) ;ADD NUMBER OF WORDS WRITTEN
4387  015144  005565  000160                ADC     P.NWRT+2(R5)
4388  015150  066565  000012  000164        ADD     P.WC(R5),P.NRD(R5) ;ADD NUMBER OF WORDS READ
4389  015156  005565  000166                ADC     P.NRD+2(R5)
4390  015162  005465  000012                NEG     P.WC(R5)            ;MAKE THE WORD COUNT NEGITIVE
4391  015166  013737  001452  177776        MOV     RKPRI,PS            ;LOCK RK06 INTERRUPTS
4392  015174  005737  001320                TST     BWAITQ              ;CHECK IF COMMANDS WAITING FOR BUFFER
4393  015200  001011                        BNE     59$                 ;YES, ENQUEUE COMMAND IN BUFFER WAIT QUEUE
4394  015202  004037  022440                JSR     R0,BUFFAL           ;GO ALLOCATE BUFFER
4395  015206  015224                        59$                         ;NOT ENOUGH ROOM, ENQUEUE IN BUFFER WAIT QUEUE
4396  015210  005037  177776                CLR     PS                  ;ALLOW RK06 INTERRUPTS
4397  015214  004737  023056                JSR     PC,LDDATA           ;LOAD BUFFER
```

```
4398  015220  000137  013662                    JMP     ASN3$           ;GO ISSUE COMMAND
4399
4400  015224  004037  051406          59$:      JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN BUFFER
4401  015230  001320                  BWAITQ                  ;  WAIT QUEUE
4402  015232  000207          60$:    RTS     PC              ;RETURN
4403
4404  015234  132765  000020  000210  WRTVER: BITB    #BIT4,P.ASSN(R5) ;CHECK IF WRITE PACK IN PROGRESS
4405  015242  001001                  BNE     1$              ;YES, CONTINUE
4406  015244  000000                  HALT                    ;  **** PROGRAM PROBLEM
4407  015246  032765  000200  000014  1$:     BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
4408  015254  001410                  BEQ     WVO$            ;YES, CHECK FOR DROP DRIVE
4409  015256  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;CLEAR ISSUE WRITE BEFORE WRITE CHECK
4410  015264  112765  000131  000001          MOVB    #WRTCHK,P.CMND(R5)      ;RESTORE WRITE CHK CMD
4411  015272  000137  013662                  JMP     ASN3$           ;GO ISSUE WRITE CHECK
4412
4413  015276  032765  002000  000014  WVO$:   BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
4414  015304  001402                  BEQ     10$             ;NO, CONTINUE
4415  015306  000137  016030                  JMP     30$             ;YES, TERMINATE PACK WRITING
4416
4417  015312  132765  000040  000210  10$:    BITB    #BIT5,P.ASSN(R5) ;CHECK IF END OF PACK
4418  015320  001402                  BEQ     15$             ;NO, CONTINUE
4419  015322  000137  016052                  JMP     WV2$            ;YES, RETURN BUFFER
4420  015326  052765  000200  000014  15$:    BIS     #W.WCK,P.PRST(R5)       ;SET ISSUE WRITE BEFORE WRT CHK
4421  015334  122765  000140  000001          CMPB    #RELEAS,P.CMND(R5)      ;SEE IF LAST CMD WAS RELEASE
4422  015342  001452                  BEQ     3$              ;BR IF YES,ELSE FILL PARAM BLOCK
4423  015344  112737  177777  001506          MOVB    #-1,I.ISRL      ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
4424  015352  012762  000300  000000          MOV     #INTR,RKCS1(R2) ;GO SCAN FOR DRIVE INTERRUPTS
4425  015360  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
4426  015364  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
4427  015372  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5) ;STORE PREVIOUS CYLINDER
4428  015400  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5) ;STORE PREVIOUS TRACK AND SECTOR
4429  015406  116565  000122  000144          MOVB    P.RDPT(R5),P.LDPT(R5) ;STORE PREVIOUS DATA PATTERN
4430  015414  016565  000130  000142          MOV     P.RWC(R5),P.LWC(R5) ;STORE PREVIOUS WORD COUNT
4431  015422  016565  000026  000004          MOV     P.DTS(R5),P.SECT(R5) ;LOAD NEW SECTOR AND TRACK
4432  015430  016565  000026  000126          MOV     P.DTS(R5),P.RSEC(R5)
4433  015436  016565  000030  000002          MOV     P.DCYL(R5),P.CYLN(R5) ;LOAD NEW CYLINDER
4434  015444  016565  000030  000124          MOV     P.DCYL(R5),P.RCYL(R5)
4435  015452  016565  000132  000010          MOV     P.RBAL(R5),P.BALO(R5) ;REINITIALIZE BUS ADDRESS FOR POSSIBLE
4436                                                  ;  BAD SECTOR RECOVERY ON LAST COMMAND
4437  015460  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)      ;SEE IF LAST CMD WAS WRT CHK
4438  015466  001404                  BEQ     4$              ;BR IF YES
4439  015470  112765  000131  000001  3$:     MOVB    #WRTCHK,P.CMND(R5)      ;ELSE SET TO DO WRT DATA (W.WCK=1)
4440  015476  000410                  BR      5$
4441  015500  042765  000200  000014  4$:     BIC     #W.WCK,P.PRST(R5)
4442  015506  112765  000140  000001          MOVB    #RELEAS,P.CMND(R5)      ;RELEAS AFTER WRT CHK
4443  015514  000137  013662                  JMP     ASN3$           ;GO RELEASE
4444  015520  016546  000002          5$:     MOV     P.CYLN(R5),-(SP) ;STORE CYLINDER ADDRESS ON STACK
4445  015524  006316                  ASL     (SP)            ;MULTIPLY CYLINDER BY 3
4446  015526  066516  000002                  ADD     P.CYLN(R5),(SP)
4447  015532  005046                  CLR     -(SP)           ;MAKE ROOM ON STACK FOR TRACK
4448  015534  116516  000005                  MOVB    P.TRCK(R5),(SP) ;STORE TRACK ON STACK
4449  015540  061666  000002                  ADD     (SP),2(SP)      ;ADD TRACK
4450  015544  012716  000026                  MOV     #22.,(SP)       ;STORE 22 ON STACK (MULTIPLIER)
4451  015550  004737  054714                  JSR     PC,$MULT        ;CALCULATE NUMBER OF SECTORS TRANSFERRED
4452  015554  012616                  MOV     (SP)+,(SP)      ;THROW AWAY MOST SIGNIFICANT BITS
4453  015556  005046                  CLR     -(SP)           ;CLEAR LOCATION FOR SECTOR
```

```
 4454  015560  116516  000004                    MOVB    P.SECT(R5),(SP) ;STORE SECTOR COUNT
 4455  015564  061666  000002                    ADD     (SP),2(SP)      ;DETERMINE TOTAL NUMBER OF SECTORS TRANSFERRED
 4456  015570  004737  027074                    JSR     PC,GETNUM       ;GET CORRECT LAST CYL NUM
 4457  015574  013716  027160                    MOV     HOLD,(SP)       ;STORE NUMBER OF SECTORS ON STACK
 4458  015600  166616  000002                    SUB     2(SP),(SP)      ;DETERMINE NUMBER OF SECTORS LEFT ON
 4459                                                                    ;  ON PACK TO BE WRITTEN
 4460  015604  016546  000130                    MOV     P.RWC(R5),-(SP) ;STORE WORD COUNT
 4461  015610  011665  000012                    MOV     (SP),P.WC(R5)
 4462  015614  005416                            NEG     (SP)            ;MAKE IT POSITIVE
 4463  015616  116616  000001                    MOVB    1(SP),(SP)      ;KEEP SECTOR COUNT
 4464  015622  105066  000001                    CLRB    1(SP)           ;CLEAR MOST SIGNIFICANT BITS
 4465  015626  021666  000002                    CMP     (SP),2(SP)      ;CHECK IF PACK OVERRUN
 4466  015632  103413                            BLO     20$             ;NO, GO ISSUE COMMAND
 4467  015634  001407                            BEQ     19$             ;CHECK IF NO NEW WORD COUNT NEEDED
 4468  015636  005016                            CLR     (SP)            ;MAKE ROOM FOR NEW WORD COUNT
 4469  015640  116666  000002  000001            MOVB    2(SP),1(SP)     ;MOVE IN NUMBER OF SECTORS LEFT
 4470  015646  005416                            NEG     (SP)            ;MAKE WORD COUNT NEGATIVE
 4471  015650  011665  000012                    MOV     (SP),P.WC(R5)   ;LOAD NEW WORD COUNT
 4472  015654  152765  000040  000210  19$:      BISB    #BIT5,P.ASSN(R5) ;SET END OF PACK TRANSFER
 4473
 4474  015662  062706  000006        20$:      ADD     #6,SP           ;ADJUST STACK
 4475  015666  016546  000012                    MOV     P.WC(R5),-(SP)  ;STORE PRESENT WORD COUNT
 4476  015672  005416                            NEG     (SP)            ;MAKE IT POSITIVE
 4477  015674  062765  000002  000152            ADD     #2,P.NODR(R5)   ;INCREMENT NUMBER OF ORDERS
 4478  015702  061665  000156                    ADD     (SP),P.NWRT(R5) ;ADD NUMBER OF WORDS WRITTEN
 4479  015706  005565  000160                    ADC     P.NWRT+2(R5)
 4480  015712  062665  000164                    ADD     (SP)+,P.NRD(R5) ;ADD NUMBER OF WORDS READ
 4481  015716  005565  000166                    ADC     P.NRD+2(R5)
 4482  015722  105265  000122                    INCB    P.RDPT(R5)      ;GENERATE NEXT DATA PATTERN TO BE WRITTEN
 4483  015726  142765  000360  000122            BICB    #360,P.RDPT(R5) ;KEEP 4 LEAST SIGNIFICANT BITS
 4484  015734  116565  000122  000121            MOVB    P.RDPT(R5),P.DPAT(R5) ;STORE DATA PATTERN USED
 4485  015742  013737  001452  177776            MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 4486  015750  132765  000040  000210            BITB    #BIT5,P.ASSN(R5) ;CHECK IF WRITE TO END OF PACK
 4487  015756  001014                            BNE     25$             ;YES, RELEASE BUFFER
 4488  015760  005737  001320                    TST     BWAITQ          ;CHECK IF BUFFER WAIT QUEUE EMPTY
 4489  015764  001011                            BNE     25$             ;NO, RELEASE BUFFER
 4490  015766  016565  000012  000130            MOV     P.WC(R5),P.RWC(R5) ;STORE CURRENT GENERATED WORD COUNT
 4491  015774  005037  177776                    CLR     PS              ;ALLOW RK06 INTERRUPTS
 4492  016000  004737  023056                    JSR     PC,LDDATA       ;LOAD NEW RANDOM DATA
 4493  016004  000137  013662                    JMP     ASN3$           ;GO ISSUE COMMAND
 4494
 4495  016010  004737  022642        25$:      JSR     PC,BUFREL       ;RELEASE BUFFER FOR OTHER DRIVES
 4496  016014  004037  051406                    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON BUFFER WAIT QUEUE
 4497  016020  001320                            BWAITQ
 4498  016022  004737  022342                    JSR     PC,GETBUF       ;GET NEW BUFFER
 4499  016026  000207                            RTS     PC              ;RETURN
 4500
 4501  016030  105065  000210        30$:      CLRB    P.ASSN(R5)      ;CLEAR ASSIGNMENT CODE
 4502  016034  004737  022642                    JSR     PC,BUFREL       ;RELEASE BUFFER
 4503  016040  104401  056260        WV1$:     TYPE    ,OPR008         ;TYPE "OPERATOR INITIATED DROP DRIVE"
 4504  016044  004737  013004                    JSR     PC,DROP         ;GO DROP DRIVE
 4505  016050  000444                            BR      WV5$            ;GO ALLOCATE BUFFER
 4506
 4507  016052  105065  000210        WV2$:     CLRB    P.ASSN(R5)      ;CLEAR ASSIGNMENT FLAGS
 4508  016056  116537  000000  056357            MOVB    P.DRVN(R5),OPR011 ;GET DRIVE NUMBER
 4509  016064  152737  000060  056357            BISB    #60,OPR011      ;MAKE IT ASCII
```

```
 4510  016072  104401  056340                TYPE    ,OPR010         ;TYPE "DRIVE NUMBER N UNDER TEST"
 4511  016076  104401  056103                TYPE    ,OPR003
 4512  016102  004737  027616                JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
 4513  016106  004737  043670                JSR     PC,PRITIM       ;PRINT TIME
 4514  016112  104401  001165                TYPE    ,$CRLF
 4515  016116  010346                        MOV     R3,-(SP)        ;STORE R3 ON STACK
 4516  016120  010446                        MOV     R4,-(SP)        ;STORE R4 ON STACK
 4517  016122  010503                        MOV     R5,R3           ;LOAD PARAMETER BLOCK BASE
 4518  016124  062703  000152                ADD     #P.NODR,R3      ;CALCULATE BEGINNING OF STATISTICS
 4519  016130  010504                        MOV     R5,R4           ;LOAD PARAMETER BLOCK BASE
 4520  016132  062704  000210                ADD     #P.ASSN,R4      ;CALCULATE END OF STATISTICS
 4521  016136  005023                7$:     CLR     (R3)+           ;CLEAR STATISTICS BEFORE STARTING TEST
 4522  016140  020304                        CMP     R3,R4           ;CHECK IF FINISHED
 4523  016142  001375                        BNE     7$              ;NO, CONTINUE
 4524  016144  012604                        MOV     (SP)+,R4        ;RESTORE R4
 4525  016146  012603                        MOV     (SP)+,R3        ;RESTORE R3
 4526  016150  004737  022642                JSR     PC,BUFREL       ;RELEASE BUFFER
 4527  016154  004037  051406                JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN DRIVE
 4528  016160  001314                        AVAILQ                  ;   AVAILIABLE QUEUE
 4529  016162  004737  022342        WV5$:   JSR     PC,GETBUF       ;GET NEW BUFFER
 4530  016166  000207                WV9$:   RTS     PC              ;RETURN
```

```
                                        .SBTTL  ALTER DRIVE PARAMETERS
                                        
016170  010446                  PARM:   MOV     R4,-(SP)        ;STORE R4 ON STACK
016172  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
016174  116537  000000  057533  PO$:    MOVB    P.DRVN(R5),PAR001 ;LOAD DRIVE NUMBER FOR PRINT OUT
016202  152737  000060  057533          BISB    #60,PAR001      ;MAKE IT ASCII
016210  104401  057507                  TYPE    ,PAR000         ;TYPE "PARAMETERS FOR DRIVE N"
016214  004737  020410                  JSR     PC,DTYPE        ;INPUT DRIVE TYPE
016220  104401  057540          2$:     TYPE    ,PAR002         ;TYPE "CYLINDER MAX,MIN"
016224  004037  020340                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
016230  016346                          5$                      ;COMMA DETECTED
016232  016362                          6$                      ;CARRIAGE RETURN DETECTED
016234  020200                          P1$                     ;CONTROL Z <↑Z> DETECTED
016236  016174                          PO$                     ;CONTROL C <↑C> DETECTED
016240  010346                  5$:     MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
016242  004737  054146                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
016246  016330                          4$                      ;ERROR RETURN
016250  012637  001602                  MOV     (SP)+,PARCYL    ;STORE MAX CYLINDER ENTRY
016254  004737  027074                  JSR     PC,GET↑NUM      ;GET CORRECT LAST CYL NUM
016260  023737  027154  001602          CMP     LST↑CYL,PARCYL  ;CHECK IF LESS OR EQUAL TO
                                                                ;  MAXIMUM CYLINDER ADDRESS
016266  103420                          BLO     4$              ;NO, TRY AGAIN
016270  010304                          MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
016272  105714                  64$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
016274  001435                          BEQ     7$              ;YES, EXIT
016276  122427  000054                  CMPB    (R4)+,#',       ;CHECK FOR COMMA
016302  001373                          BNE     64$             ;NO, GO TEST NEXT CHARACTER
016304  010446                  3$:     MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
016306  004737  054146                  JSR     PC,OCTBIN       ;CONVERT TO BINARY
016312  016330                          4$                      ;ERROR RETURN
016314  012637  001604                  MOV     (SP)+,PARCYL+2  ;STORE MIN CYLINDER ENTRY
016320  023737  001602  001604          CMP     PARCYL,PARCYL+2 ;CHECK IF LEGAL ENTRY
016326  103022                          BHIS    8$              ;YES, LOAD MAX AND MIN CYLINDER
016330                          4$:
016330  010337  016336                  MOV     R3,65$          ;LOAD BUFFER ADDRESS
016334  104401                          TYPE                    ;TYPE RECEIVED INPUT
016336  000000                  65$:    .WORD   0               ;BUFFER ADDRESS
016340  104401  001164                  TYPE    ,$QUES          ;TYPE QUESTION MARK
016344  000725                          BR      2$              ;TRY AGAIN

016346  013737  027154  001602  5$:     MOV     LSTCYL,PARCYL   ;LOAD MAX CYLINDER = 632/1456
016354  010304                          MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
016356  105724                          TSTB    (R4)+           ;ADJUST R4
016360  000751                          BR      3$              ;GET MINIMUM CYLINDER

016362  013737  027154  001602  6$:     MOV     LSTCYL,PARCYL   ;LOAD MAX CYLINDER = 632/1456
016370  005037  001604          7$:     CLR     PARCYL+2        ;LOAD MIN CYLINDER = 0
016374  013765  001602  000070  8$:     MOV     PARCYL,P.MXCL(R5) ;LOAD MAX CYLINDER
016402  013765  001604  000066          MOV     PARCYL+2,P.MNCL(R5) ;LOAD MIN CYLINDER

016410  104401  057563          10$:    TYPE    ,PAR003         ;TYPE "TRACK MAX,MIN"
016414  004037  020340                  JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
016420  016532                          13$                     ;COMMA DETECTED
016422  016546                          14$                     ;CARRIAGE RETURN DETECTED
016424  020212                          P2$                     ;CONTROL Z <↑Z> DETECTED
016426  016174                          PO$                     ;CONTROL C <↑C> DETECTED
```

```
4587  016430  010346                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
4588  016432  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
4589  016436  016514                        12$                     ;ERROR RETURN
4590  016440  012637  001606                MOV     (SP)+,PARTRK    ;STORE MAX TRACK ENTRY
4591  016444  022737  000002  001606        CMP     #D.MXTR,PARTRK  ;CHECK IF LESS OR EQUAL TO
4592                                                                ;  MAXIMUM TRACK ADDRESS
4593  016452  103420                        BLO     12$             ;NO, TRY AGAIN
4594  016454  010304                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
4595  016456  105714          66$:          TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
4596  016460  001435                        BEQ     15$             ;YES, EXIT
4597  016462  122427  000054                CMPB    (R4)+,#',       ;CHECK FOR COMMA
4598  016466  001373                        BNE     66$             ;NO, GO TEST NEXT CHARACTER
4599  016470  010446          11$:          MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
4600  016472  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
4601  016476  016514                        12$                     ;ERROR RETURN
4602  016500  012637  001610                MOV     (SP)+,PARTRK+2  ;STORE MIN TRACK ENTRY
4603  016504  023737  001606  001610        CMP     PARTRK,PARTRK+2 ;CHECK IF LEGAL ENTRY
4604  016512  103022                        BHIS    16$             ;YES, LOAD MAX AND MIN TRACK
4605  016514          12$:
4606  016514  010337  016522                MOV     R3,67$          ;LOAD BUFFER ADDRESS
4607  016520  104401                        TYPE                    ;TYPE RECEIVED INPUT
4608  016522  000000          67$:          .WORD   0               ;BUFFER ADDRESS
4609  016524  104401  001164                TYPE    $QUES           ;TYPE QUESTION MARK
4610  016530  000727                        BR      10$             ;TRY AGAIN
4611
4612  016532  012737  000002  001606  13$:  MOV     #D.MXTR,PARTRK  ;LOAD MAX TRACK = 2
4613  016540  010304                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
4614  016542  105724                        TSTB    (R4)+           ;ADJUST R4
4615  016544  000751                        BR      11$             ;GET MINIMUM TRACK
4616
4617  016546  012737  000002  001606  14$:  MOV     #D.MXTR,PARTRK  ;LOAD MAX TRACK = 2
4618  016554  005037  001610          15$:  CLR     PARTRK+2        ;LOAD MIN TRACK = 0
4619  016560  113765  001606  000073  16$:  MOVB    PARTRK,P.MXTR(R5) ;LOAD MAX TRACK
4620  016566  113765  001610  000072        MOVB    PARTRK+2,P.MNTR(R5) ;LOAD MIN TRACK
4621
4622  016574  104401  057603          18$:  TYPE    ,PAR004         ;TYPE "SECTOR MAX,MIN"
4623  016600  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4624  016604  016716                        21$                     ;COMMA DETECTED
4625  016606  016732                        22$                     ;CARRIAGE RETURN DETECTED
4626  016610  020224                        P3$                     ;CONTROL Z (↑Z) DETECTED
4627  016612  016174                        P0$                     ;CONTROL C (↑C) DETECTED
4628  016614  010346                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
4629  016616  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
4630  016622  016700                        20$                     ;ERROR RETURN
4631  016624  012637  001612                MOV     (SP)+,PARSEC    ;STORE MAX SECTOR ENTRY
4632  016630  022737  000025  001612        CMP     #D.MXSC,PARSEC  ;CHECK IF LESS OR EQUAL TO
4633                                                                ;  MAXIMUM SECTOR ADDRESS
4634  016636  103420                        BLO     20$             ;NO, TRY AGAIN
4635  016640  010304                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
4636  016642  105714          68$:          TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
4637  016644  001435                        BEQ     23$             ;YES, EXIT
4638  016646  122427  000054                CMPB    (R4)+,#',       ;CHECK FOR COMMA
4639  016652  001373                        BNE     68$             ;NO, GO TEST NEXT CHARACTER
4640  016654  010446          19$:          MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
4641  016656  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
4642  016662  016700                        20$                     ;ERROR RETURN
```

```
4643  016664  012637  001614                MOV     (SP)+,PARSEC+2  ;STORE MIN SECTOR ENTRY
4644  016670  023737  001612  001614        CMP     PARSEC,PARSEC+2 ;CHECK IF LEGAL ENTRY
4645  016676  103022                        BHIS    24$             ;YES, LOAD MAX AND MIN SECTOR
4646  016700                          20$:
4647  016700  010337  016706                MOV     R3,69$          ;LOAD BUFFER ADDRESS
4648  016704  104401                        TYPE                    ;TYPE RECEIVED INPUT
4649  016706  000000                  69$:  .WORD   0               ;BUFFER ADDRESS
4650  016710  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
4651  016714  000727                        BR      18$             ;TRY AGAIN
4652
4653  016716  012737  000025  001612  21$:  MOV     #D.MXSC,PARSEC  ;LOAD MAX SECTOR = 21
4654  016724  010304                        MOV     R3,R4           ;STORE R3 IN R4 FOR LINE SCAN
4655  016726  105724                        TSTB    (R4)+           ;ADJUST R4
4656  016730  000751                        BR      19$             ;GET MINIMUM SECTOR
4657
4658  016732  012737  000025  001612  22$:  MOV     #D.MXSC,PARSEC  ;LOAD MAX SECTOR = 21
4659  016740  005037  001614          23$:  CLR     PARSEC+2        ;LOAD MIN SECTOR = 0
4660  016744  113765  001612  000075  24$:  MOVB    PARSEC,P.MXSC(R5) ;LOAD MAX SECTOR
4661  016752  113765  001614  000074        MOVB    PARSEC+2,P.MNSC(R5) ;LOAD MIN SECTOR
4662
4663  016760  104401  057624          26$:  TYPE    ,PAR005         ;TYPE "READ/WRITE RATIO ="
4664  016764  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4665  016770  017020                        27$                     ;COMMA DETECTED
4666  016772  017044                        29$                     ;CARRIAGE RETURN DETECTED
4667  016774  020236                        P4$                     ;CONTROL Z (↑Z) DETECTED
4668  016776  016174                        P0$                     ;CONTROL C (↑C) DETECTED
4669  017000  010346                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
4670  017002  004737  054146                JSR     PC,OCTBIN       ;CONVERT BINARY
4671  017006  017020                        27$                     ;ERROR RETURN
4672  017010  012604                        MOV     (SP)+,R4        ;STORE INPUT
4673  017012  022704  000010                CMP     #10,R4          ;CHECK IF 0-7
4674  017016  101007                        BHI     28$             ;YES, LOAD READ/WRITE RATIO
4675  017020                          27$:
4676  017020  010337  017026                MOV     R3,70$          ;LOAD BUFFER ADDRESS
4677  017024  104401                        TYPE                    ;TYPE RECEIVED INPUT
4678  017026  000000                  70$:  .WORD   0               ;BUFFER ADDRESS
4679  017030  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
4680  017034  000751                        BR      26$             ;TRY AGAIN
4681
4682  017036  110465  000076          28$:  MOVB    R4,P.RATE(R5)   ;LOAD READ/WRITE RATIO
4683  017042  000403                        BR      30$             ;ASK "WRITE CHECK AFTER WRITE ="
4684
4685  017044  112765  000003  000076  29$:  MOVB    #D.RATE,P.RATE(R5) ;LOAD DEFAULT READ/WRITE
4686
4687  017052  104401  057647          30$:  TYPE    ,PAR006         ;TYPE "WRITE CHECK AFTER WRITE ="
4688  017056  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4689  017062  017110                        31$                     ;COMMA DETECTED
4690  017064  017126                        32$                     ;CARRIAGE RETURN DETECTED
4691  017066  020244                        P5$                     ;CONTROL Z (↑Z) DETECTED
4692  017070  016174                        P0$                     ;CONTROL C (↑C) DETECTED
4693  017072  122713  000131                CMPB    #'Y,(R3)        ;CHECK IF YES
4694  017076  001013                        BNE     32$             ;NO, LOAD DEFAULT
4695  017100  112765  177777  000077        MOVB    #-1,P.AWCK(R5)  ;SET↑ WRITE CHECK AFTER WRITE
4696  017106  000411                        BR      34$             ;GET CORRECTABLE READ ERROR
4697                                                                ;   THRESHOLD
4698
```

```
                                31$:
017110                                  MOV     R3,71$          ;LOAD BUFFER ADDRESS
017110  010337  017116                  TYPE                    ;TYPE RECEIVED INPUT
017114  104401                  71$:    .WORD   0               ;BUFFER ADDRESS
017116  000000                  71$:    TYPE    ,$QUES          ;TYPE QUESTION MARK
017120  104401  001164                  BR      30$             ;TRY AGAIN
017124  000752
                                32$:    CLRB    P.AWCK(R5)      ;CLEAR WRITE CHECK AFTER WRITE
017126  105065  000077
                                34$:    TYPE    ,PAR007         ;TYPE "CORRECTABLE READ ERROR
017132  104401  057701                                          ;   THRESHOLD
                                        JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017136  004037  020340                  35$                     ;COMMA DETECTED
017142  017170                          36$                     ;CARRIAGE RETURN DETECTED
017144  017206                          P6$                     ;CONTROL Z <↑Z> DETECTED
017146  020250                          P0$                     ;CONTROL C <↑C> DETECTED
017150  016174                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
017152  010346                          JSR     PC,DECBIN       ;CONVERT TO BINARY
017154  004737  054302                  35$                     ;ERROR RETURN
017160  017170                          MOV     (SP)+,P.CERT(R5) ;LOAD CORRECTABLE READ ERROR
017162  012665  000100                                          ;   THRESHOLD
                                        BPL     38$             ;IF NOT NEGATIVE, GET UNCORRECTABLE
017166  100012                                                  ;   READ ERROR THRESHOLD

                                35$:
017170                                  MOV     R3,72$          ;LOAD BUFFER ADDRESS
017170  010337  017176                  TYPE                    ;TYPE RECEIVED INPUT
017174  104401                  72$:    .WORD   0               ;BUFFER ADDRESS
017176  000000                  72$:    TYPE    ,$QUES          ;TYPE QUESTION MARK
017200  104401  001164                  BR      34$             ;TRY AGAIN
017204  000752
017206  012765  000012  000100  36$:    MOV     #D.CERT,P.CERT(R5) ;LOAD DEFAULT CORRECTABLE READ
                                                                ;   ERROR THRESHOLD

                                38$:    TYPE    ,PAR008         ;TYPE "UNCORRECTABLE READ ERROR
017214  104401  057744                                          ;   THRESHOLD ="
                                        JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017220  004037  020340                  39$                     ;COMMA DETECTED
017224  017252                          40$                     ;CARRIAGE RETURN DETECTED
017226  017270                          P7$                     ;CONTROL Z <↑Z> DETECTED
017230  020256                          P0$                     ;CONTROL C <↑C> DETECTED
017232  016174                          MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
017234  010346                          JSR     PC,DECBIN       ;CONVERT TO BINARY
017236  004737  054302                  39$                     ;ERROR RETURN
017242  017252                          MOV     (SP)+,P.UERT(R5) ;LOAD UNCORRECTABLE READ ERROR
017244  012665  000102                                          ;   THRESHOLD
                                        BPL     42$             ;IF NOT NEGATIVE, GET SEEK
017250  100012                                                  ;   ERROR THRESHOLD

                                39$:
017252                                  MOV     R3,73$          ;LOAD BUFFER ADDRESS
017252  010337  017260                  TYPE                    ;TYPE RECEIVED INPUT
017256  104401                  73$:    .WORD   0               ;BUFFER ADDRESS
017260  000000                  73$:    TYPE    ,$QUES          ;TYPE QUESTION MARK
017262  104401  001164                  BR      38$             ;TRY AGAIN
017266  000752
017270  012765  000005  000102  40$:    MOV     #D.UERT,P.UERT(R5) ;LOAD DEFAULT UNCORRECTABLE READ
```

```
                                                                ;  ERROR THRESHOLD
                                                                ;
017276  104401  060011        42$:    TYPE    ,PAR009         ;TYPE "SEEK ERROR THRESHOLD ="
017302  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017306  017334                        43$                     ;COMMA DETECTED
017310  017352                        44$                     ;CARRIAGE RETURN DETECTED
017312  020264                        P8$                     ;CONTROL Z (↑Z) DETECTED
017314  016174                        P0$                     ;CONTROL C (↑C) DETECTED
017316  010346                MOV     R3,-(SP)                ;STORE BUFFER ADDRESS
017320  004737  054302        JSR     PC,DECBIN               ;CONVERT TO BINARY
017324  017334                43$                             ;ERROR RETURN
017326  012665  000104        MOV     (SP)+,P.SKET(R5)        ;LOAD SEEK ERROR THRESHOLD
017332  100012                BPL     46$                     ;CHECK IF INHIBIT ERROR CORRECTION
                                                                ;  AND RETRY
017334                        43$:
017334  010337  017342        MOV     R3,74$                  ;LOAD BUFFER ADDRESS
017340  104401                TYPE                            ;TYPE RECEIVED INPUT
017342  000000                74$:    .WORD   0               ;BUFFER ADDRESS
017344  104401  001164        TYPE    ,$QUES                  ;TYPE QUESTION MARK
017350  000752                BR      42$                     ;TRY AGAIN
                                                                ;
017352  012765  000005  000104  44$:    MOV     #D.SKET,P.SKET(R5) ;LOAD DEFAULT SEEK ERROR THRESHOLD
                                                                ;
017360  104401  060040        46$:    TYPE    ,PAR010         ;TYPE "INHIBIT ERROR CORRECTION
                                                                ;  AND RETRY
017364  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017370  017416                        47$                     ;COMMA DETECTED
017372  017434                        48$                     ;CARRIAGE RETURN DETECTED
017374  020272                        P9$                     ;CONTROL Z (↑Z) DETECTED
017376  016174                        P0$                     ;CONTROL C (↑C) DETECTED
017400  122713  000131                CMPB    #'Y,(R3)        ;CHECK IF YES
017404  001013                BNE     48$                     ;LOAD DEFAULT
017406  112765  177777  000120        MOVB    #-1,P.IERC(R5)  ;SET INHIBIT ERROR CORRECTION
017414  000411                BR      50$                     ;GET COMMAND COUNT THRESHOLD
                                                                ;
017416                        47$:
017416  010337  017424                MOV     R3,75$          ;LOAD BUFFER ADDRESS
017422  104401                TYPE                            ;TYPE RECEIVED INPUT
017424  000000                75$:    .WORD   0               ;BUFFER ADDRESS
017426  104401  001164        TYPE    ,$QUES                  ;TYPE QUESTION MARK
017432  000752                BR      46$                     ;TRY AGAIN
                                                                ;
017434  105065  000120        48$:    CLRB    P.IERC(R5)      ;CLEAR INHIBIT ERROR CORRECTION
                                                                ;
017440  104401  060105        50$:    TYPE    ,PAR011         ;TYPE "OPERATION COUNT THRESHOLD*65K ="
017444  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
017450  017514                        51$                     ;COMMA DETECTED
017452  017532                        52$                     ;CARRIAGE RETURN DETECTED
017454  020276                        P10$                    ;CONTROL Z (↑Z) DETECTED
017456  016174                        P0$                     ;CONTROL C (↑C) DETECTED
017460  010346                MOV     R3,-(SP)                ;STORE BUFFER ADDRESS
017462  004737  054302        JSR     PC,DECBIN               ;CONVERT TO BINARY
017466  017514                        51$                     ;ERROR RETURN
017470  012665  000110        MOV     (SP)+,P.MXCD+2(R5)      ;STORE NUMBER OF 65K BLOCKS
017474  001407                BEQ     51$                     ;IF ZERO, TRY AGAIN
017476  100406                BMI     51$                     ;IF MINUS, TRY AGAIN
```

```
4811  017500  005365  000110                DEC     P.MXCD+2(R5)        ;DECREMENT BY 1
4812  017504  012765  177777  000106        MOV     #-1,P.MXCD(R5)
4813  017512  000415                        BR      54$                 ;GET WORD TRANSFERREC THRESHOLD
4814
4815  017514                        51$:
4816  017514  010337  017522                MOV     R3,76$              ;LOAD BUFFER ADDRESS
4817  017520  104401                        TYPE                        ;TYPE RECEIVED INPUT
4818  017522  000000                76$:    .WORD   0                   ;BUFFER ADDRESS
4819  017524  104401  001164                TYPE    $QUES               ;TYPE QUESTION MARK
4820  017530  000743                        BR      50$                 ;TRY AGAIN
4821
4822  017532  012765  077777  000110  52$:  MOV     #D.CTRH,P.MXCD+2(R5) ;LOAD DEFAULT COMMAND THRESHOLD
4823  017540  012765  177770  000106        MOV     #D.CTRL,P.MXCD(R5)
4824
4825  017546  104401  060145        54$:    TYPE    ,PAR012             ;TYPE "WORD TRNSFERRED THRESHOLD*520K ="
4826  017552  004037  020340                JSR     R0,TSTDEF           ;CHECK FOR DEFAULT PARAMETERS
4827  017556  017654                        55$                         ;COMMA DETECTED
4828  017560  017672                        56$                         ;CARRIAGE RETURN DETECTED
4829  017562  020312                        P11$                        ;CONTROL Z (↑Z) DETECTED
4830  017564  016174                        P0$                         ;CONTROL C (↑C) DETECTED
4831  017566  010346                        MOV     R3,-(SP)            ;STORE BUFFER ADDRESS
4832  017570  004737  054302                JSR     PC,DECBIN           ;CONVERT TO BINARY
4833  017574  017654                        55$                         ;ERROR RETURN
4834  017576  012665  000112                MOV     (SP)+,P.MXWT(R5)    ;STORE NUMBER OF 520K BLOCKS
4835  017602  001424                        BEQ     55$                 ;IF ZERO, TRY AGAIN
4836  017604  100423                        BMI     55$                 ;IF NEGATIVE, TRY AGAIN
4837  017606  005065  000114                CLR     P.MXWT+2(R5)        ;CLEAR HIGH ORDER BITS
4838  017612  006165  000112                ROL     P.MXWT(R5)          ;SHIFT 4 BITS LEFT
4839  017616  006165  000114                ROL     P.MXWT+2(R5)
4840  017622  006165  000112                ROL     P.MXWT(R5)
4841  017626  006165  000114                ROL     P.MXWT+2(R5)
4842  017632  006165  000112                ROL     P.MXWT(R5)
4843  017636  006165  000114                ROL     P.MXWT+2(R5)
4844  017642  006165  000112                ROL     P.MXWT(R5)
4845  017646  006165  000114                ROL     P.MXWT+2(R5)
4846  017652  000415                        BR      60$                 ;CHECK FOR SAMPLED COMPARES
4847
4848  017654                        55$:
4849  017654  010337  017662                MOV     R3,77$              ;LOAD BUFFER ADDRESS
4850  017660  104401                        TYPE                        ;TYPE RECEIVED INPUT
4851  017662  000000                77$:    .WORD   0                   ;BUFFER ADDRESS
4852  017664  104401  001164                TYPE    $QUES               ;TYPE QUESTION MARK
4853  017670  000726                        BR      54$                 ;TRY AGAIN
4854
4855  017672  012765  077777  000114  56$:  MOV     #D.WTHI,P.MXWT+2(R5) ;LOAD DEFAULT WORD TRANSFER
4856  017700  012765  177770  000112        MOV     #D.WTLO,P.MXWT(R5)      ;  THRESHOLD
4857
4858  017706  005737  001362        60$:    TST     SOFCMP              ;CHECK IF ANY SOFTWARE COMPARES ARE TO BE MADE
4859  017712  001430                        BEQ     PREXAR              ;NO, DETERMINE TRACK EXCLUSION
4860  017714  104401  060443                TYPE    ,PAR021             ;TYPE "SAMPLED COMPARES ="
4861  017720  004037  020340                JSR     R0,TSTDEF           ;CHECK FOR DEFAULT PARAMETERS
4862  017724  017752                        61$                         ;COMMA DETECTED
4863  017726  017770                        62$                         ;CARRIAGE RETURN DETECTED
4864  017730  020326                        P12$                        ;CONTROL Z (↑Z) DETECTED
4865  017732  016174                        P0$                         ;CONTROL C (↑C) DETECTED
4866  017734  122713  000116                CMPB    #'N,(R3)            ;CHECK IF NO
```

```
4867  017740  001013                        BNE     62$             ;NO, MAKE SAMPLED COMPARES
4868  017742  112765  177777  000116        MOVB    #-1,P.SMPL(R5)  ;DO NOT MAKE SAMPLED COMPARES
4869  017750  000411                        BR      PREXAR          ;DETERMINE TRACK EXCLUSION
4870
4871  017752                        61$:
4872  017752  010337  017760                MOV     R3,78$          ;LOAD BUFFER ADDRESS
4873  017756  104401                        TYPE                    ;TYPE RECEIVED INPUT
4874  017760  000000                78$:    .WORD   0               ;BUFFER ADDRESS
4875  017762  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
4876  017766  000747                        BR      60$             ;TRY AGAIN
4877
4878  017770  105065  000116        62$:    CLRB    P.SMPL(R5)      ;MAKE SAMPLED COMPARES
4879
4880  017774  104401  060300        PREXAR: TYPE    ,PAR019         ;TYPE "CHANGE EXCUDED PACK AREA ="
4881  020000  104401  060252                TYPE    ,PAR018
4882  020004  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
4883  020010  020032                        1$                      ;COMMA DETECTED
4884  020012  020172                        50$                     ;CARRIAGE RETURN DETECTED
4885  020014  020172                        50$                     ;CONTROL Z <↑Z> DETECTED
4886  020016  016174                        P0$                     ;CONTROL C <↑C> DETECTED
4887  020020  122713  000131                CMPB    #'Y,(R3)        ;CHECK IF YES
4888  020024  001411                        BEQ     2$              ;YES, LOAD NEW TRACK EXCULSIONS
4889  020026  000137  020172                JMP     50$             ;NO, RETURN
4890
4891  020032                        1$:
4892  020032  010337  020040                MOV     R3,64$          ;LOAD BUFFER ADDRESS
4893  020036  104401                        TYPE                    ;TYPE RECEIVED INPUT
4894  020040  000000                64$:    .WORD   0               ;BUFFER ADDRESS
4895  020042  104401  001164                TYPE    ,$QUES          ;TYPE QUESTION MARK
4896  020046  000752                        BR      PREXAR          ;TRY AGAIN
4897
4898  020050  010146                2$:     MOV     R1,-(SP)        ;STORE R1 ON STACK
4899  020052  010501                        MOV     R5,R1           ;STORE PARMETER BLOCK ADDRESS
4900  020054  062701  000232                ADD     #P.EXAR,R1      ;CALCULATE EXCLUSION AREA
4901  020060  012704  000012                MOV     #10.,R4         ;LOAD COUNT
4902  020064  005021                5$:     CLR     (R1)+           ;CLEAR EXCLUSION AREA
4903  020066  005304                        DEC     R4              ;DECREMENT COUNT
4904  020070  001375                        BNE     5$              ;CHECK IF DONE
4905  020072  162701  000024                SUB     #20.,R1         ;REINITIALIZE POINTER
4906  020076  004037  020506                JSR     R0,LDEXAR       ;GET FIRST EXCLUDED AREA
4907  020102  060212                        PAR013
4908  020104  020170                        45$                     ;DEFAULT RETURN
4909  020106  020162                        40$                     ;CONTROL C <↑C> RETURN
4910  020110  004037  020506                JSR     R0,LDEXAR       ;GET SECOND EXCLUDED AREA
4911  020114  060220                        PAR014
4912  020116  020170                        45$                     ;DEFAULT RETURN
4913  020120  020162                        40$                     ;CONTROL C <↑C> RETURN
4914  020122  004037  020506                JSR     R0,LDEXAR       ;GET THIRD EXCLUDED AREA
4915  020126  060227                        PAR015
4916  020130  020170                        45$                     ;DEFAULT RETURN
4917  020132  020162                        40$                     ;CONTROL C <↑C> RETURN
4918  020134  004037  020506                JSR     R0,LDEXAR       ;GET FOURTH EXCLUDED AREA
4919  020140  060235                        PAR016
4920  020142  020170                        45$                     ;DEFAULT RETURN
4921  020144  020162                        40$                     ;CONTROL C <↑C> RETURN
4922  020146  004037  020506                JSR     R0,LDEXAR       ;GET FIFTH EXCLUDED AREA
```

```
 4923  020152  060244                          PAR017
 4924  020154  020170                          45$                     ;DEFAULT RETURN
 4925  020156  020162                          40$                     ;CONTROL C (↑C) RETURN
 4926  020160  000403                          BR      45$             ;RETURN
 4927
 4928  020162  012601                  40$:    MOV     (SP)+,R1        ;RESTORE R1
 4929  020164  000137  016174                  JMP     P0$             ;START AGAIN
 4930
 4931  020170  012601                  45$:    MOV     (SP)+,R1        ;RESTORE R1
 4932  020172  012603                  50$:    MOV     (SP)+,R3        ;RESTORE R3
 4933  020174  012604                          MOV     (SP)+,R4        ;RESTORE R4
 4934  020176  000207                          RTS     PC              ;RETURN
 4935
 4936  020200  013765  027154  000070  P1$:    MOV     LSTCYL,P.MXCL(R5) ;LOAD DEFAULT MAXIMUM CYLINDER = 632/1456
 4937  020206  005065  000066                  CLR     P.MNCL(R5)      ;LOAD DEFAULT MINIMUM CYLINDER
 4938  020212  112765  000002  000073  P2$:    MOVB    #D.MXTR,P.MXTR(R5) ;LOAD DEFAULT MAXIMUM TRACK = 2
 4939  020220  105065  000072                  CLRB    P.MNTR(R5)      ;LOAD DEFAULT MINIMUM TRACK
 4940  020224  112765  000025  000075  P3$:    MOVB    #D.MXSC,P.MXSC(R5) ;LOAD DEFAULT MAXIMUM SECTOR = 21
 4941  020232  105065  000074                  CLRB    P.MNSC(R5)      ;LOAD DEFAULT MINIMUM SECTOR
 4942  020236  112765  000003  000076  P4$:    MOVB    #D.RATE,P.RATE(R5) ;LOAD READ/WRITE RATIO
 4943  020244  105065  000077          P5$:    CLRB    P.AWCK(R5)      ;LOAD WRITE CHECK AFTER WRITE
 4944  020250  012765  000012  000100  P6$:    MOV     #D.CERT,P.CERT(R5) ;LOAD CORRECTABLE READ ERROR THRESHOLD
 4945  020256  012765  000005  000102  P7$:    MOV     #D.UERT,P.UERT(R5) ;LOAD UNCORRECTABLE READ ERROR THRESHOLD
 4946  020264  012765  000005  000104  P8$:    MOV     #D.SKET,P.SKET(R5) ;LOAD SEEK ERROR THRESHOLD
 4947  020272  105065  000120          P9$:    CLRB    P.IERC(R5)      ;CLEAR INHIBIT ERROR CORRECTION
 4948  020276  012765  077777  000110  P10$:   MOV     #D.CTRH,P.MXCD+2(R5) ;LOAD COMMAND THRESHOLD
 4949  020304  012765  177770  000106          MOV     #D.CTRL,P.MXCD(R5)
 4950  020312  012765  077777  000114  P11$:   MOV     #D.WTHI,P.MXWT+2(R5) ;LOAD WORD TRANSFERED THRESHOLD
 4951  020320  012765  177770  000112          MOV     #D.WTLO,P.MXWT(R5)
 4952  020326  105065  000116          P12$:   CLRB    P.SMPL(R5)      ;MAKE SAMPLED COMPARES
 4953
 4954  020332  012603                          MOV     (SP)+,R3        ;RESTORE R3
 4955  020334  012604                          MOV     (SP)+,R4        ;RESTORE R4
 4956  020336  000207                          RTS     PC              ;RETURN
```

```
                        .SBTTL  TEST DEFAULT PARAMETERS ROUTINE
                        ;;*****************************************************************
                        ;;*
                        ;;*      THIS ROUTINE WILL READ A LINE OF ASCII TEXT AND CHECK
                        ;;*      IF PARAMETER OR REST OF PARAMETERS WILL BE DEFAULTED.
                        ;;*      IT RETURNS WITH R3 POINTING TO THE ADDRESS OF THE BUFFER.
                        ;;*
                        ;;*CALL  JSR     R0,TSTDEF
                        ;;*      <ADDRESS OF COMMA DETECTION RETURN>
                        ;;*      <ADDRESS OF CARRIAGE RETURN DETECTION RETURN>
                        ;;*      <ADDRESS OF CONTROL Z (↑Z) DETECTION RETURN>
                        ;;*      <ADDRESS OF CONTROL C (↑C) DETECTION RETURN>
                        ;;*      RETURN
                        ;;*
                        ;;*****************************************************************

020340  104403          TSTDEF: RDLIN                   ;GET RESPONSE
020342  012603                  MOV     (SP)+,R3        ;STORE BUFFER ADDRESS
020344  121327  000054          CMPB    (R3),#',        ;CHECK IF COMMA
020350  001415                  BEQ     10$             ;YES, RETURN
020352  005720                  TST     (R0)+           ;ADJUST R0
020354  105713                  TSTB    (R3)            ;CHECK IF CARRIAGE RETURN
020356  001412                  BEQ     10$             ;YES, RETURN
020360  005720                  TST     (R0)+           ;ADJUST R0
020362  122713  000032          CMPB    #32,(R3)        ;CHECK IF CONTROL Z (↑Z)
020366  001406                  BEQ     10$             ;YES, RETURN
020370  005720                  TST     (R0)+           ;ADJUST R0
020372  122713  000003          CMPB    #3,(R3)         ;CHECK IF CONTROL C (↑C)
020376  001402                  BEQ     10$             ;YES, RETURN
020400  005720                  TST     (R0)+           ;ADJUST R0
020402  000200                  RTS     R0              ;RETURN

020404  011000          10$:    MOV     (R0),R0         ;STORE RETURN ADDRESS
020406  000200                  RTS     R0              ;RETURN

020410  104401  060466  DTYPE:  TYPE    ,PAR022         ;TYPE "TESTING RK06? Y OR N"
020414  004037  020340          JSR     R0,TSTDEF       ;GET INPUT
020420  020444                  1$                      ;COMMA
020422  020444                  1$                      ;CR
020424  020444                  1$                      ;CONT Z
020426  020444                  1$                      ;CONT C

020430  121327  000131          CMPB    (R3),#'Y        ;SEE IF RK06
020434  001406                  BEQ     2$              ;BR IF YES
020436  121327  000116          CMPB    (R3),#'N        ;SEE IF RK07
020442  001412                  BEQ     3$              ;BR IF YES

020444  104401  056042  1$:     TYPE    ,ILLCMD         ;ELSE ERROR
020450  000757                  BR      DTYPE           ;TRY AGAIN

020452  142765  000004  000007  2$:     BICB    #B.CDT,P.CS1H(R5)       ;RK06
020460  012737  000632  027154          MOV     #632,LSTCYL             ;LAST CYLINDER ADDRESS
020466  000207                  RTS     PC
020470  152765  000004  000007  3$:     BISB    #B.CDT,P.CS1H(R5)       ;RK07
020476  012737  001456  027154          MOV     #1456,LSTCYL            ;LAST CYL ADDRESS
```

```
 5013  020504  000207                        RTS     PC
 5014
 5015
```

```
                                .SBTTL  LOAD DISK EXCLUDED AREAS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL ACCEPT TTY INPUT FOR EXCLUDED AREAS AND
                                ;*      LOAD APPROPIATE IDENTIFICATION NUMBERS IN THE EXCLUDED AREAS OF
                                ;*      THE PARAMETER BLOCK.  THE FOLLOWING THREE FORMATS ARE
                                ;*      ACCEPTED BY THIS ROUTINE:
                                ;*
                                ;*              CYL
                                ;*              CYL,TRK
                                ;*              CYL1,TRK1,CYL2,TRK2
                                ;*
                                ;*      THE FOLLOWING FORMULA IS USED:
                                ;*
                                ;*              CYL1*66+TRK1*22+1 = BEGINNING OF EXCLUDED AREA
                                ;*              CYL2*66+TRK2*22+1 = END OF EXCLUDED AREA
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R1              ADDRESS IN EXCLUDED AREA
                                ;*      R3              ADDRESS OF INPUT STRING
                                ;*      R4              ADDRESS IN INPUT STRING
                                ;*      R5              PARAMETER BLOCK ADDRESS
                                ;*
                                ;*CALL   JSR     R0,LDEXAR
                                ;*      <ADDRESS OF MESSAGE TO PRINT>
                                ;*      <ADDDRESS FOR CARRIAGE RETURN OR CONTROL Z <↑Z> DETECTION>
                                ;*      <ADDRESS FOR CONTROL C <↑C> DETECTION>
                                ;*      RETURN
                                ;*
                                ;;*****************************************************************

020506  012037  020514        LDEXAR: MOV     (R0)+,2$        ;LOAD NUMBER OF EXCULSION AREA

020512  104401                1$:     TYPE                    ;TYPE "NTH EXCLUDED AREA ="
020514  000000                2$:     .WORD   0
020516  104401  060252                TYPE    .PAR018
020522  004037  020340                JSR     R0,TSTDEF       ;CHECK FOR DEFAULT PARAMETERS
020526  021132                        30$                     ;COMMA DETECTED
020530  021154                        45$                     ;CARRIAGE RETURN DETECTED
020532  021154                        45$                     ;CONTROL Z <↑Z> DETECTED
020534  021152                        40$                     ;CONTROL C <↑C> DETECTED
020536  010346                        MOV     R3,-(SP)        ;STORE BUFFER ADDRESS
020540  004737  054146                JSR     PC,OCTBIN       ;CONVERT TO BINARY
020544  021132                        30$                     ;ERROR RETURN
020546  023716  027154                CMP     LSTCYL,(SP)     ;CHECK IF LEGAL CYLINDER
020552  103566                        BLO     29$             ;NO, TRY AGAIN
020554  011637  001616                MOV     (SP),EXAREA     ;MULTIPLY CYLINDER ADDRESS BY 3
020560  006316                        ASL     (SP)
020562  062637  001616                ADD     (SP)+,EXAREA
020566  010304                        MOV     R3,R4           ;STORE BEGINNING OF COMMAND STRING

020570  105714                64$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
020572  001525                        BEQ     10$             ;YES, EXIT
```

```
5072  020574  122427  000054          CMPB    (R4)+,#',       ;CHECK FOR COMMA
5073  020600  001373                  BNE     64$             ;NO, GO TEST NEXT CHARACTER
5074  020602  105714                  TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5075  020604  001552                  BEQ     30$             ;YES, TRY AGAIN
5076  020606  121427  000054          CMPB    (R4),#',        ;CHECK IF COMMA
5077  020612  001547                  BEQ     30$             ;YES, TRY AGAIN
5078  020614  010446                  MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5079  020616  004737  054146          JSR     PC,OCTBIN       ;CONVERT TO BINARY
5080  020622  021132                  30$                     ;ERROR RETURN
5081  020624  022716  000002          CMP     #2,(SP)         ;CHECK IF LEGAL TRACK
5082  020630  103537                  BLO     29$             ;TRY AGAIN
5083  020632  063716  001616          ADD     EXAREA,(SP)     ;PUT MULTIPLIER ON STACK
5084  020636  012746  000026          MOV     #22.,-(SP)      ;PUT MULTIPLIER ON STACK
5085  020642  004737  054714          JSR     PC,$MULT        ;DO MULTIPLICATION
5086  020646  005216                  INC     (SP)            ;ADD 1
5087  020650  012637  001616          MOV     (SP)+,EXAREA    ;STORE VALUE
5088  020654  005726                  TST     (SP)+           ;THROW AWAY MOST SIGNIFICANT BITS
5089
5090  020656  105714          65$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
5091  020660  001511                  BEQ     20$             ;YES, EXIT
5092  020662  122427  000054          CMPB    (R4)+,#',       ;CHECK FOR COMMA
5093  020666  001373                  BNE     65$             ;NO, GO TEST NEXT CHARACTER
5094  020670  105714                  TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5095  020672  001517                  BEQ     30$             ;YES, TRY AGAIN
5096  020674  121427  000054          CMPB    (R4),#',        ;CHECK IF COMMA
5097  020700  001514                  BEQ     30$             ;YES, TRY AGAIN
5098  020702  010446                  MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5099  020704  004737  054146          JSR     PC,OCTBIN       ;CONVERT TO BINARY
5100  020710  021132                  30$                     ;ERROR RETURN
5101  020712  023716  027154          CMP     LSTCYL,(SP)     ;CHECK IF LEGAL CYLINDER
5102  020716  103504                  BLO     29$             ;NO, TRY AGAIN
5103  020720  011637  001620          MOV     (SP),EXAREA+2   ;MULTIPLY CYLINDER ADDRESS BY 3
5104  020724  006316                  ASL     (SP)
5105  020726  062637  001620          ADD     (SP)+,EXAREA+2
5106
5107  020732  105714          66$:    TSTB    (R4)            ;CHECK FOR CARRIAGE RETURN
5108  020734  001476                  BEQ     30$             ;YES, EXIT
5109  020736  122427  000054          CMPB    (R4)+,#',       ;CHECK FOR COMMA
5110  020742  001373                  BNE     66$             ;NO, GO TEST NEXT CHARACTER
5111  020744  105714                  TSTB    (R4)            ;CHECK IF CARRIAGE RETURN
5112  020746  001471                  BEQ     30$             ;YES, TRY AGAIN
5113  020750  121427  000054          CMPB    (R4),#',        ;CHECK IF COMMA
5114  020754  001466                  BEQ     30$             ;YES, TRY AGAIN
5115  020756  010446                  MOV     R4,-(SP)        ;STORE BUFFER ADDRESS
5116  020760  004737  054146          JSR     PC,OCTBIN       ;CONVERT TO BINARY
5117  020764  021132                  30$                     ;ERROR RETURN
5118  020766  022716  000002          CMP     #2,(SP)         ;CHECK IF LEGAL TRACK
5119  020772  103456                  BLO     29$             ;TRY AGAIN
5120  020774  063716  001620          ADD     EXAREA+2,(SP)   ;PUT MULTIPLIER ON STACK
5121  021000  012746  000026          MOV     #22.,-(SP)      ;PUT MULTIPLIER ON STACK
5122  021004  004737  054714          JSR     PC,$MULT        ;DO MULTIPLICATION
5123  021010  005216                  INC     (SP)            ;ADD 1
5124  021012  012637  001620          MOV     (SP)+,EXAREA+2  ;STORE VALUE
5125  021016  005726                  TST     (SP)+           ;THROW AWAY MOST SIGNIFICANT BITS
5126  021020  023737  001616  001620  CMP     EXAREA,EXAREA+2 ;CHECK IF LEGAL RANGE
5127  021026  103041                  BHIS    30$             ;NO, TRY AGAIN
```

```
5128  021030  013721  001616          MOV     EXAREA,(R1)+    ;LOAD EXCLUDED AREA
5129  021034  013721  001620          MOV     EXAREA+2,(R1)+
5130  021040  062700  000004          ADD     #4,R0           ;ADJUST R0
5131  021044  000200                  RTS     R0              ;RETURN
5132
5133  021046  013746  001616    10$:  MOV     EXAREA,-(SP)    ;LOAD MULTIPLIER
5134  021052  012746  000026          MOV     #22.,-(SP)      ;LOAD MULTICAND
5135  021056  004737  054714          JSR     PC,$MULT        ;CALCULATE VALUE
5136  021062  005216                  INC     (SP)            ;INCREMENT RESULTS
5137  021064  011621                  MOV     (SP),(R1)+      ;STORE EXCLUDED AREA
5138  021066  062716  000102          ADD     #66.,(SP)
5139  021072  012621                  MOV     (SP)+,(R1)+
5140  021074  005726                  TST     (SP)+           ;THROW AWAY MUST SIGNIFICANT BITS
5141  021076  062700  000004          ADD     #4,R0           ;ADJUST R0
5142  021102  000200                  RTS     R0              ;RETURN
5143
5144  021104  013721  001616    20$:  MOV     EXAREA,(R1)+    ;STORE EXCLUDEED AREA
5145  021110  062737  000026  001616  ADD     #22.,EXAREA
5146  021116  013721  001616          MOV     EXAREA,(R1)+
5147  021122  062700  000004          ADD     #4,R0           ;ADJUST R0
5148  021126  000200                  RTS     R0              ;RETURN
5149
5150  021130  005726            29$:  TST     (SP)+           ;ADJUST STACK
5151  021132  010337  021140    30$:  MOV     R3,31$          ;LOAD BUFFER ADDRESS
5152  021136  104401                  TYPE                    ;TYPE RECEIVED INPUT
5153  021140  000000            31$:  .WORD   0               ;BUFFER ADDRESS
5154  021142  104401  001164          TYPE    ,$QUES          ;TYPE QUESTION MARK
5155  021146  000137  020512          JMP     1$              ;TRY AGAIN
5156
5157  021152  005720            40$:  TST     (R0)+           ;ADJUST R0
5158  021154  011000            45$:  MOV     (R0),R0         ;STORE SPECIAL RETURN ADDRESS
5159  021156  000200                  RTS     R0              ;RETURN
```

```
                        .SBTTL  GENERATE NEW RANDOM COMMAND

                        ;;*****************************************************************
                        ;*
                        ;*      IF SWITCH 14 IS RESET, THIS ROUTINE WILL STORE THE PREVIOUS
                        ;*      COMMAND AND GENERATE A NEW RANDOM COMMAND.  A CHECK IS MADE
                        ;*      ON THE DISK ADDRESS AND THE DISK ADDRESS + THE NUMBER OF
                        ;*      SECTORS TO BE READ OR WRITTEN.  IF THE DATA TRANSFER WILL
                        ;*      CAUSE DATA TO BE READ OR WRITTEN BEYOND THE MAXIMUM PACK
                        ;*      ADDRESS OR USES ANY PART OF AN EXCLUDED AREA, A NEW DISK
                        ;*      ADDRESS WILL BE RANDOMLY GENERATED.
                        ;*
                        ;*      REGISTER        USE
                        ;*      --------        ---
                        ;*
                        ;*      R1              EXCLUSION AREA COUNT
                        ;*      R3              EXCLUSION AREA INDEX
                        ;*      R4              SECTOR POSITION
                        ;*      R5              PARAMETER BLOCK ADDRESS
                        ;*      (SP)            SECTORS TO BE TRANSFERRED
                        ;*
                        ;*
                        ;*      IF SWITCH 14 IS SET, THIS ROUTINE WILL SEEK TO THE PREVIOUS
                        ;*      DISK POSITION OR REISSUE THE LAST COMMAND DEPENDENT ON THE
                        ;*      P.SEEK FLAG IN THE PARAMETER BLOCK.
                        ;*
                        ;*
                        ;*      ROUTINES USED
                        ;*      -------------
                        ;*
                        ;*      Q.PUSH
                        ;*      $RAND
                        ;*      $MULT
                        ;*
                        ;*CALL  JSR     PC,GENERT
                        ;*      RETURN
                        ;*
                        ;;*****************************************************************

021160  005065  000146        GENERT: CLR     P.RECT(R5)      ;CLEAR ERROR FLAGS AND
021164  005065  000150                CLR     P.DSTT(R5)      ;  RETRY COUNTS
021170  005065  000212                CLR     P.ERR(R5)

021174  032777  040000  157736        BIT     #SW14,@SWR      ;CHECK IF NEW COMMAND
                                                                ;  SHOULD BE GENERATED
021202  001454                        BEQ     10$             ;YES, GENERATE NEW COMMAND
021204  105765  000211                TSTB    P.SEEK(R5)      ;CHECK IF PREVIOUS CYLINDER ISSUED
021210  001020                        BNE     1$              ;YES, SET UP DATA TRANSFER
021212  112765  177777  000211        MOVB    #-1,P.SEEK(R5)  ;INDICATE SEEK HAS BEEN ISSUED
021220  112765  000117  000001        MOVB    #SEEK,P.CMND(R5)  ;LOAD SEEK PARAMETERS
021226  016565  000136  000002        MOV     P.LCYL(R5),P.CYLN(R5)  ;LOAD PRIOR CYLINDER
021234  016565  000140  000004        MOV     P.LSEC(R5),P.SECT(R5)  ;LOAD PRIOR SECTOR AND TRACK
021242  004037  051406                JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
021246  001324                        CINITQ                  ;  COMMAND INITIATION QUEUE
021250  000207                        RTS     PC              ;RETURN
```

```
5216  021252  105065  000211          1$:     CLRB    P.SEEK(R5)          ;INDICATE DATA TRANSFER ISSUED
5217  021256  116565  000122  000121          MOVB    P.RDPT(R5),P.DPAT(R5)  ;LOAD LAST DATA PATTERN
5218  021264  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5)  ;LOAD LAST COMMAND
5219  021272  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5)  ;LOAD LAST CYLINDER ADDRESS
5220  021300  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)  ;CHECK IF WRITE CHECK
5221  021306  001003                          BNE     2$                  ;NO, DO NOT SET FLAG
5222  021310  052765  000200  000014          BIS     #W.WCK,P.PRST(R5)   ;SET ISSUE WRITE FOR WRITE CHECK
5223  021316  016565  000126  000004  2$:     MOV     P.RSEC(R5),P.SECT(R5)  ;LOAD LAST SECTOR AND TRACK
5224  021324  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5)  ;LOAD LAST WORD COUNT
5225  021332  000207                          RTS     PC                  ;RETURN
5226
5227  021334  005077  157604          10$:    CLR     @$TKS               ;LOCK OUT TTY INTERRUPTS
5228  021340  010446                          MOV     R4,-(SP)            ;STORE R4 ON STACK
5229  021342  010346                          MOV     R3,-(SP)            ;STORE R3 ON STACK
5230  021344  010146                          MOV     R1,-(SP)            ;STORE R1 ON STACK
5231  021346  010046                          MOV     R0,-(SP)            ;STORE R0 ON STACK
5232  021350  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5)  ;STORE LAST COMMAND
5233  021356  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5)  ;STORE LAST CYLINDER
5234  021364  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5)  ;STORE LAST SECTOR AND TRACK
5235  021372  016565  000130  000142          MOV     P.RWC(R5),P.LWC(R5)  ;STORE LAST WORD COUNT
5236  021400  116565  000122  000144          MOVB    P.RDPT(R5),P.LDPT(R5) ;STORE LAST PATTERN
5237  021406  004737  055030                  JSR     PC,$RAND            ;GENERATE RANDOM NUMBER
5238  021412  013746  055130                  MOV     $LONUM,-(SP)        ;LOAD MULTIPLIER
5239  021416  013746  001354                  MOV     MAXBUF,-(SP)        ;LOAD MULTIPLICAND
5240  021422  005216                          INC     (SP)                ;INCREMENT WORD COUNT MAX.
5241  021424  004737  054714                  JSR     PC,$MULT            ;CALCULATE WORD COUNT
5242  021430  005726                          TST     (SP)+               ;THROW AWAY LEAST SIGNIFICANT BITS
5243  021432  042716  100003                  BIC     #100003,(SP)        ;MAKE DIVISIBLE BY 4 AND < 32K
5244  021436  001002                          BNE     11$                 ;IF NOT ZERO USE WORD COUNT
5245  021440  012716  000004                  MOV     #4,(SP)             ;MAKE WORD COUNT 4
5246  021444  116604  000001          11$:    MOVB    1(SP),R4            ;CALCULATE NUMBER OF SECTORS TO
5247  021450  105716                          TSTB    (SP)                ;  TRANSFERRED
5248  021452  001401                          BEQ     12$                 ;IF EVEN SECTORS DO NOT MODIFY
5249  021454  005204                          INC     R4                  ;  SECTOR COUNT
5250  021456  011646                  12$:    MOV     (SP),-(SP)          ;SAVE WORD COUNT
5251  021460  005416                          NEG     (SP)                ;GENERATE NEGATIVE NUMBER
5252  021462  012665  000012                  MOV     (SP)+,P.WC(R5)      ;LOAD PRESENT WORD COUNT
5253  021466  062765  000001  000152          ADD     #1,P.NODR(R5)       ;INCREMENT NUMBER OF ORDERS
5254  021474  005565  000154                  ADC     P.NODR+2(R5)
5255  021500  113746  055126                  MOVB    $HINUM,-(SP)        ;RANDOM NUMBER FOR COMMAND
5256  021504  042716  177770                  BIC     #177770,(SP)        ;KEEP LOW 3 BITS
5257  021510  126526  000076                  CMPB    P.RATE(R5),(SP)+    ;DETERMINE IF READ OR WRITE
5258  021514  101462                          BLOS    15$                 ;ISSUE READ
5259  021516  113746  055127                  MOVB    $HINUM+1,-(SP)      ;RANDOM NUMBER FOR PATTERN
5260  021522  042716  177760                  BIC     #177760,(SP)        ;KEEP LOW 4 BITS
5261  021526  111665  000122                  MOVB    (SP),P.RDPT(R5)     ;STORE DATA PATTERN
5262  021532  112665  000121                  MOVB    (SP)+,P.DPAT(R5)
5263  021536  105765  000077                  TSTB    P.AWCK(R5)          ;CHECK IF AUTO WRITE CHECK
5264  021542  001017                          BNE     13$                 ;YES, ISSUE WRITE CHECK
5265  021544  113746  055126                  MOVB    $HINUM,-(SP)        ;RANDOM NUMBER FOR WRITE CHECK
5266  021550  042726  177707                  BIC     #177707,(SP)+       ;CHECK IF WRITE COMMAND
5267  021554  001412                          BEQ     13$                 ;YES, ISSUE WRITE CHECK
5268  021556  112765  000123  000123          MOVB    #WRDATA,P.RCMD(R5)  ;ISSUE WRITE DATA
5269  021564  062665  000156                  ADD     (SP)+,P.NWRT(R5)    ;ADD NUMBER OF WORDS TO BE WRITTEN
5270  021570  005565  000160                  ADC     P.NWRT+2(R5)
5271  021574  005565  000162                  ADC     P.NWRT+4(R5)
```

```
5272  021600  000441                          BR      18$
5273
5274  021602  112765  000131  000123  13$:    MOVB    #WRTCHK,P.RCMD(R5)  ;ISSUE WRITE CHECK
5275  021610  052765  000200  000014          BIS     #W.WCK,P.PRST(R5)  ;SET NO WRITE FOR WRITE CHECK
5276  021616  062765  000001  000152          ADD     #1,P.NODR(R5)   ;INCREMENT NUMBER OF COMMAND ISSUED
5277  021624  005565  000154                  ADC     P.NODR+2(R5)
5278  021630  061665  000156                  ADD     (SP),P.NWRT(R5) ;ADD NUMBER OF WORDS TO BE WRITTEN
5279  021634  005565  000160                  ADC     P.NWRT+2(R5)
5280  021640  005565  000162                  ADC     P.NWRT+4(R5)
5281  021644  062665  000164                  ADD     (SP)+,P.NRD(R5) ;ADD NUMBER OF WORDS READ
5282  021650  005565  000166                  ADC     P.NRD+2(R5)
5283  021654  005565  000170                  ADC     P.NRD+4(R5)
5284  021660  000411                          BR      18$
5285
5286  021662  112765  000121  000123  15$:    MOVB    #RDDATA,P.RCMD(R5)  ;ISSUE READ DATA
5287  021670  062665  000164                  ADD     (SP)+,P.NRD(R5) ;ADD NUMBER OF WORDS READ
5288  021674  005565  000166                  ADC     P.NRD+2(R5)
5289  021700  005565  000170                  ADC     P.NRD+4(R5)
5290  021704  116565  000123  000001  18$:    MOVB    P.RCMD(R5),P.CMND(R5)  ;LOAD COMMAND
5291  021712  012700  000100                  MOV     #100,R0         ;LOAD REGENERATE COUNT
5292
5293  021716  010503                  20$:    MOV     R5,R3           ;LOAD INDEX
5294  021720  062703  000232                  ADD     #P.EXAR,R3      ;CALCULATE EXCLUDED AREA INDEX
5295  021724  012701  000005                  MOV     #5,R1           ;LOAD BAD TRACK COUNT
5296  021730  004737  055030                  JSR     PC,$RAND        ;GENERATE RANDOM NUMBER
5297  021734  016546  000070                  MOV     P.MXCL(R5),-(SP)  ;STORE MAXIMUM CYLINDER
5298  021740  166516  000066                  SUB     P.MNCL(R5),(SP) ;CALCULATE DIFFERENCE
5299  021744  001406                          BEQ     21$             ;IF 0, SKIP MULTIPLY
5300  021746  005216                          INC     (SP)            ;INCREMENT DIFFERENCE
5301  021750  013746  055130                  MOV     $LONUM,-(SP)    ;PUT MULTICAND ON STACK
5302  021754  004737  054714                  JSR     PC,$MULT        ;CALCULATE CYLINDER ADDRESS
5303  021760  005726                          TST     (SP)+           ;THROW AWAY LEASE SIGNIFICANT BITS
5304
5305  021762  066516  000066          21$:    ADD     P.MNCL(R5),(SP) ;CALCULATE CYLINDER
5306  021766  011665  000124                  MOV     (SP),P.RCYL(R5) ;STORE CYLINDER
5307  021772  116537  000073  001572          MOVB    P.MXTR(R5),TMPTRK  ;STORE MAXIMUM TRACK
5308  022000  116537  000072  001574          MOVB    P.MNTR(R5),TMPTRK+2  ;STORE MINIMUM TRACK
5309  022006  163737  001574  001572          SUB     TMPTRK+2,TMPTRK ;CALCULATE PERFERENCE
5310  022014  001415                          BEQ     22$             ;IF=0, SKIP MULTIPLY
5311  022016  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
5312  022020  113716  055127                  MOVB    $HINUM+1,(SP)   ;PUT IN MULTIPLIER ON STACK
5313  022024  013746  001572                  MOV     TMPTRK,-(SP)    ;PUT TRACK NUMBER ON STACK
5314  022030  005216                          INC     (SP)            ;INCREMENT DIFFERENCE
5315  022032  004737  054714                  JSR     PC,$MULT        ;CALCULATE TRACK
5316  022036  116637  000001  001572          MOVB    1(SP),TMPTRK    ;STORE TRACK
5317  022044  062706  000004                  ADD     #4,SP           ;ADJUST STACK
5318
5319  022050  063737  001574  001572  22$:    ADD     TMPTRK+2,TMPTRK ;CALCULATE TRACK
5320  022056  113765  001572  000127          MOVB    TMPTRK,P.RTRK(R5)  ;STORE TRACK NUMBER
5321  022064  116537  000075  001576          MOVB    P.MXSC(R5),TMPSEC  ;STORE MAXIMUM SECTOR
5322  022072  116537  000074  001600          MOVB    P.MNSC(R5),TMPSEC+2  ;STORE MINIMUM SECTOR
5323  022100  163737  001600  001576          SUB     TMPSEC+2,TMPSEC ;CALCULATE DIFFERENCE
5324  022106  001415                          BEQ     23$             ;IF=0, SKIP MULTIPLY
5325  022110  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
5326  022112  113716  055126                  MOVB    $HINUM,(SP)     ;PUT IN MULTIPLIER
5327  022116  013746  001576                  MOV     TMPSEC,-(SP)    ;PUT SECTOR NUMBER ON STACK
```

```
5328  022122  005216                          INC     (SP)            ;INCREMENT DIFFERENCE
5329  022124  004737  054714                  JSR     PC,$MULT        ;CALCULATE SECTOR
5330  022130  116637  000001  001576          MOVB    1(SP),TMPSEC    ;STORE SECTOR
5331  022136  062706  000004                  ADD     #4,SP           ;ADJUST STACK
5332
5333  022142  063737  001600  001576  23$:    ADD     TMPSEC+2,TMPSEC ;CALCULATE SECTOR
5334  022150  113765  001576  000126          MOVB    TMPSEC,P.RSEC(R5)  ;STORE SECTOR NUMBER
5335  022156  006316                          ASL     (SP)            ;MULTIPLY CYLINDER BY 3
5336  022160  066516  000124                  ADD     P.RCYL(R5),(SP)
5337  022164  063716  001572                  ADD     TMPTRK,(SP)     ;ADD TRACK
5338  022170  012746  000026                  MOV     #22.,-(SP)      ;MULTIPLICAND - 22 SECTORS/TRACK
5339  022174  004737  054714                  JSR     PC,$MULT
5340  022200  012616                          MOV     (SP)+,(SP)      ;THROW AWAY MOST SIGNIFICANT BITS
5341  022202  063716  001576                  ADD     TMPSEC,(SP)     ;ADD SECTOR
5342  022206  005216                          INC     (SP)            ;CONSTANT FOR BAD TRACKS
5343  022210  061604                          ADD     (SP),R4         ;ADD NUMBER OF SECTORS
5344  022212  004737  027074                  JSR     PC,GETNUM
5345  022216  023704  027162                  CMP     HOLD1,R4        ;CHK IF DATA XFER EXEEDS DISK
5346  022222  103421                          BLO     40$             ;CALCULATE NEW DISK ADDRESS
5347
5348  022224  005713          25$:    TST     (R3)                    ;CHECK IF BAD TRACK INTERVAL
5349                                                                  ;  EXIST
5350  022226  001407                          BEQ     30$             ;NO, ISSUE COMMAND
5351  022230  021623                          CMP     (SP),(R3)+      ;CHECK IF IN BAD TRACK INTERVAL
5352  022232  103402                          BLO     26$             ;NO, CHECK NEXT BAD TRACK 1
5353  022234  020413                          CMP     R4,(R3)
5354  022236  103413                          BLO     40$             ;YES, CALCULATE NEW DISK ADDRESS
5355  022240  005723          26$:    TST     (R3)+                   ;ADJUST R3
5356  022242  005301                          DEC     R1              ;DECREMENT BAD BLOCK COUNT
5357  022244  001367                          BNE     25$             ;IF NOT 0 GET NEXT BAD BLOCK
5358
5359  022246  005726          30$:    TST     (SP)+                   ;ADJUST STACK
5360  022250  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5)  ;LOAD CYLINDER
5361  022256  016565  000126  000004          MOV     P.RSEC(R5),P.SECT(R5)  ;LOAD TRACK AND SECTOR
5362  022264  000416                          BR      45$             ;RETURN
5363
5364  022266  162604          40$:    SUB     (SP)+,R4                ;GET NUMBER OF SECTOR TRANSFERRED
5365  022270  005300                          DEC     R0              ;DECREMENT REGENERATE COUNT
5366  022272  001402                          BEQ     41$             ;DROP DRIVE CANNOT GENERATE
5367                                                                  ;  RANDOM DISK ADDRESS
5368  022274  000137  021716                  JMP     20$             ;GET ANOTHER DISK ADDRESS
5369
5370  022300  013737  001452  177776  41$:    MOV     RKPRI,PS        ;LOCK OUT RK06 AND TTY INERRUPTS
5371  022306  104401  060307                  TYPE    ,PAR020         ;TYPE "UNABLE TO GENERATE NEW COMMAND"
5372  022312  004737  013004                  JSR     PC,DROP         ;DROP DRIVE
5373  022316  005037  177776                  CLR     PS              ;ALLOW RK06 INTERRUPTS
5374
5375  022322  012600          45$:    MOV     (SP)+,R0                ;RESTORE R0
5376  022324  012601                  MOV     (SP)+,R1                ;RESTORE R1
5377  022326  012603                  MOV     (SP)+,R3                ;RESTORE R3
5378  022330  012604                  MOV     (SP)+,R4                ;RESTORE R4
5379  022332  012777  000100  156604          MOV     #IE,@$TKS       ;ALLOW TTY INTERRUPTS
5380  022340  000207                  RTS     PC                      ;RETURN
```

```
5381                                    .SBTTL  GET AVAILIBLE BUFFER ROUTINE
5382
5383   022342  013746  177776           GETBUF: MOV     PS,-(SP)        ;STORE PSW ON STACK
5384   022346  013737  001452  177776           MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
5385   022354  013705  001320                   MOV     BWAITQ,R5       ;CHECK IF COMMAND WAITING FOR BUFFER
5386   022360  001424                           BEQ     10$             ;RETURN IF QUEUE IS EMPTY
5387   022362  004037  022440           1$:     JSR     R0,BUFFAL       ;ALLOCATE BUFFER
5388   022366  022432                           10$                     ;NOT ENOUGH MEMORY RETURN
5389   022370  004037  051466                   JSR     R0,Q.POP        ;GET NEXT PARMETER BLOCK FROM
5390   022374  001320                           BWAITQ                  ;   BUFFER WAIT QUEUE
5391   022376  005726                           TST     (SP)+           ;ADJUST STACK
5392   022400  005037  177776                   CLR     PS              ;ALLOW RK06 INTERRUPTS
5393   022404  004737  023056                   JSR     PC,LDDATA       ;LOAD BUFFER
5394   022410  004037  051406                   JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
5395   022414  001324                           CINITQ                  ;   COMMAND INITIATION QUEUE
5396   022416  013737  001452  177776           MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
5397   022424  013705  001320                   MOV     BWAITQ,R5       ;CHECK IF COMMAND WAITING FOR BUFFER
5398   022430  001354                           BNE     1$              ;YES, TRY TO ALLOCATE BUFFER
5399
5400   022432  012637  177776           10$:    MOV     (SP)+,PS        ;RESTORE PSW
5401   022436  000207                           RTS     PC              ;RETURN
```

```
                                .SBTTL  BUFFER ALLOCATION ROUTINE

                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL TRY TO ALLOCATE A MAIN MEMORY BUFFER
                                ;*      IF MEMORY IS AVAILABLE.  (CALLED BY PRIORITY 7)
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R1              WORD COUNT # 2
                                ;*      R3              FREE BLOCK TABLE INDEX
                                ;*      R4              FREE BLOCK TABLE COUNT
                                ;*      R5              PARAMETER BLOCK ADDRESS
                                ;*
                                ;*CALL  JSR     R0,BUFFAL
                                ;*      <ADDRESS OF BUFFER MEMORY NOT AVAILABLE ROUTINE>
                                ;*      RETURN  <BUFFER ALLOCATED>
                                ;*
                                ;*****************************************************************

022440  010446          BUFFAL: MOV     R4,-(SP)        ;STORE R4 ON STACK
022442  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
022444  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
022446  016501  000012          MOV     P.WC(R5),R1     ;STORE WORD COUNT
022452  010165  000130          MOV     R1,P.RWC(R5)    ;STORE WORD COUNT
022456  005401                  NEG     R1              ;GET ACTUAL WORD COUNT
022460  006301                  ASL     R1              ;MULTIPLY WORD COUNT BY 2
022462  013704  001670          MOV     FBLKC,R4        ;STORE FREE BLOCK TABLE ADDRESS
022466  001420                  BEQ     2$              ;RETURN, IF NO BLOCKS AVAILABLE
022470  012703  001672          MOV     #FBLKT,R3       ;LOAD ADDRESS OF FREE BLOCK TABLE
022474  022704  000011          CMP     #9.,R4          ;HANG IF FREE BLOCK COUNT IS GREATER
022500  101001                  BHI     1$              ;  OR EQUAL TO 9
022502  000000                  HALT

022504  012346          1$:     MOV     (R3)+,-(SP)     ;STORE BEGINNING OF FREE BLOCK
022506  060116                  ADD     R1,(SP)         ;ADD WORD COUNT
022510  103411                  BCS     5$              ;CHECK IF ADDRESS OVERFLOW
022512  021316                  CMP     (R3),(SP)       ;CHECK IF ENOUGH ROOM
022514  001423                  BEQ     15$             ;YES, ELIMINATE BLOCK
022516  101010                  BHI     10$             ;YES, SHORTEN BLOCK
022520  005723                  TST     (R3)+           ;ADJUST R3
022522  005726                  TST     (SP)+           ;ADJUST STACK
022524  005304                  DEC     R4              ;DECREMENT FREE BLOCK COUNT
022526  001366                  BNE     1$              ;EXAMINE NEXT FREE BLOCK
022530  011000          2$:     MOV     (R0),R0         ;LOAD RETURN ADDRESS
022532  000437                  BR      2S$             ;RETURN

022534  005726          5$:     TST     (SP)+           ;ADJUST STACK
022536  000774                  BR      2$              ;RETURN

022540  014365  000132  10$:     MOV     -(R3),P.RBAL(R5) ;LOAD BUS ADDRESS
022544  011365  000010          MOV     (R3),P.BALO(R5) ;LOAD BUS ADDRESS
022550  012613                  MOV     (SP)+,(R3)      ;LOAD NEW BASE
022552  005720                  TST     (R0)+           ;ADJUST RETURN
022554  112765  177777  000145  MOVB    #-1,P.BUFF(R5)  ;INDICATE THAT BUFFER HAS BEEN ASSIGNED
```

```
5458  022562  000423                          BR      25$             ;RETURN
5459
5460  022564  010301                  15$:    MOV     R3,R1           ;STORE WORD COUNT
5461  022566  014365  000132                  MOV     -(R3),P.RBAL(R5)
5462  022572  011365  000010                  MOV     (R3),P.BALO(R5) ;LOAD BUS ADDRESS
5463  022576  005726                          TST     (SP)+           ;ADJUST STACK
5464  022600  005721                          TST     (R1)+           ;ADJUST R1
5465  022602  005720                          TST     (R0)+           ;ADJUST RETURN
5466  022604  112765  177777  000145          MOVB    #-1,P.BUFF(R5)  ;INDICATE THAT BUFFER HAS BEEN ASSIGNED
5467  022612  005337  001670                  DEC     FBLKC           ;DECREMENT FREE BLOCK COUNT
5468  022616  005304                          DEC     R4              ;DECREMENT REMAIN BLOCK COUNT
5469  022620  001404                          BEQ     25$             ;IF 0, RETURN
5470
5471  022622  012123                  20$:    MOV     (R1)+,(R3)+     ;TRANSFER TWO WORDS
5472  022624  012123                          MOV     (R1)+,(R3)+
5473  022626  005304                          DEC     R4              ;DECREMENT REMAINDING BLOCKS TO BE ADJUSTED
5474  022630  001374                          BNE     20$             ;IF NOT 0, TRANSFER NEXT TWO WORDS
5475  022632  012601                  25$:    MOV     (SP)+,R1        ;RESTORE R1
5476  022634  012603                          MOV     (SP)+,R3        ;RESTORE R3
5477  022636  012604                          MOV     (SP)+,R4        ;RESTORE R4
5478  022640  000200                          RTS     R0              ;RETURN
5479
```

```
                                .SBTTL  BUFFER RELEASE ROUTINE

                                ;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL RELEASE BUFFER USED.
                                ;*      (MUST BE CALLED IN PRIORITY 7)
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R0              NEW END OF FREE BLOCK TABLE
                                ;*      R1              LAST ADDRESS OF RELEASED BUFFER
                                ;*      R3              FREE BLOCK INDEX
                                ;*      R4              FREE BLOCK COUNT
                                ;*      R5              PARAMETER BLOCK ADDRESS
                                ;*      (SP)            BEGINNING ADDRESS OF RELEASED BLOCK
                                ;*
                                ;*CALL  JSR     PC,BUFREL
                                ;*      RETURN
                                ;*
                                ;*****************************************************************

022642  010446          BUFREL: MOV     R4,-(SP)        ;STORE R4 ON STACK
022644  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
022646  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
022650  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
022652  105765  000145          TSTB    P.BUFF(R5)      ;CHECK IF BUFFER ASSIGNED
022656  001472                  BEQ     2$              ;NO, RETURN
022660  105065  000145          CLRB    P.BUFF(R5)      ;CLEAR BUFFER ALLOCATED
022664  016501  000130          MOV     P.RWC(R5),R1    ;STORE ALLOCATED WORD COUNT
022670  005401                  NEG     R1              ;GET ACTUAL WORD COUNT
022672  006301                  ASL     R1              ;MULTIPLY BY 2
022674  016546  000132          MOV     P.RBAL(R5),-(SP)  ;STORE BUFFER ADDRESS
022700  061601                  ADD     (SP),R1         ;CALCULATE FINAL ADDRESS
022702  012703  001672          MOV     #FBLKT,R3       ;LOAD FREE BLOCK TABLE ADDRESS
022706  013704  001670          MOV     FBLKC,R4        ;STORE FREE BLOCK COUNT
022712  001424                  BEQ     8$              ;ADD FREE BLOCK

022714  022301          1$:     CMP     (R3)+,R1        ;COMPARE END OF BUFFER AND FREE BLOCK
022716  001427                  BEQ     10$             ;EQUAL MERGE BLOCKS
022720  101005                  BHI     5$              ;LESS THEN ADD FREE BLOCK
022722  022316                  CMP     (R3)+,(SP)      ;COMPARE BEGINNING OF BUFFER
                                                        ;   AND END OF FREE BLOCK
022724  001426                  BEQ     15$             ;EQUAL MERGE
022726  005304                  DEC     R4              ;DECREMENT FREE BLOCK COUNT
022730  001371                  BNE     1$              ;GO TO NEXT FREE BLOCK
022732  000414                  BR      8$              ;GO ADD BLOCK

022734  010446          5$:     MOV     R4,-(SP)        ;PUT REMAINING COUNT ON STACK
022736  006316                  ASL     (SP)            ;MULTIPLY BY 4
022740  006316                  ASL     (SP)
022742  005743                  TST     -(R3)           ;DECREMENT R3 BY 2
022744  062603                  ADD     (SP)+,R3        ;CALCULATE OLD END OF FREE BLOCK TABLE
022746  010300                  MOV     R3,R0
022750  062700  000004          ADD     #4,R0           ;CALCULATE NEW END OF FREE BLOCK TABLE
022754  014340          6$:     MOV     -(R3),-(R0)     ;MOVE TABLE
```

```
5536  022756  014340                        MOV     -(R3),-(R0)
5537  022760  005304                        DEC     R4              ;DECREMENT FREE BLOCK COUNT
5538  022762  001374                        BNE     6$              ;IF NOT ZERO, COUNT
5539
5540  022764  005237  001670          8$:   INC     FBLKC           ;INCREMENT FREE BLOCK COUNT
5541  022770  012623                        MOV     (SP)+,(R3)+     ;STORE BEGINNING OF FREE BLOCK
5542  022772  010123                        MOV     R1,(R3)+        ;STORE END OF FREE BLOCK
5543  022774  000423                        BR      25$             ;RETURN
5544
5545  022776  012643                  10$:  MOV     (SP)+,-(R3)     ;LOAD BEGINNING OF FREE BLOCK
5546  023000  000421                        BR      25$             ;RETURN
5547
5548  023002  005726                  15$:  TST     (SP)+           ;ADJUST STACK
5549  023004  005304                        DEC     R4              ;DECREMENT FREE BLOCKS LEFT
5550  023006  001402                        BEQ     16$             ;NO MORE FREE BLOCK ADJUST END
5551  023010  021301                        CMP     (R3),R1         ;CHECK IF RELEASED BLOCK END=
5552                                                                ;  FREE BLOCK BEGINNING
5553  023012  001402                        BEQ     20$             ;YES, MERGE BLOCKS
5554  023014  010143                  16$:  MOV     R1,-(R3)        ;LOAD END OF FREE BLOCK
5555  023016  000412                        BR      25$             ;RETURN
5556
5557  023020  010300                  20$:  MOV     R3,R0           ;STORE R3 FOR BUFFER MOVEMENT
5558  023022  005743                        TST     -(R3)           ;ADJUST R3
5559  023024  005720                        TST     (R0)+           ;ADJUST R0
5560  023026  006304                        ASL     R4              ;MULTIPLY COUNT BY TWO
5561  023030  005304                        DEC     R4              ;DECREMENT TRANSFER COUNT
5562  023032  012023                  21$:  MOV     (R0)+,(R3)+     ;MOVE TABLE ENTRY
5563  023034  005304                        DEC     R4              ;DECREMENT COUNT
5564  023036  001375                        BNE     21$             ;IF NOT 0, CONTINUE MOVE
5565  023040  005337  001670                DEC     FBLKC           ;DECREMENT FREE BLOCK COUNT
5566
5567  023044  012600                  25$:  MOV     (SP)+,R0        ;RESTORE R0
5568  023046  012601                        MOV     (SP)+,R1        ;RESTORE R1
5569  023050  012603                        MOV     (SP)+,R3        ;RESTORE R3
5570  023052  012604                        MOV     (SP)+,R4        ;RESTORE R4
5571  023054  000207                        RTS     PC              ;RETURN
5572
```

```
5573                                    .SBTTL  LOAD DATA BUFFER
5574
5575                                    ;;*****************************************************************
5576                                    ;*
5577                                    ;*      THE REGISTERS BE USED AS FOLLOWS:
5578                                    ;*
5579                                    ;*      REGISTER                USE
5580                                    ;*      --------                ---
5581                                    ;*
5582                                    ;*         R5           ADDRESS OF PARAMETER BLOCK
5583                                    ;*         R4           BUFFER ADDRESS
5584                                    ;*         R3           PATTERN ADDRESS
5585                                    ;*         R2           WORDS LEFT IN SECTOR
5586                                    ;*         R1           WORD COUNT
5587                                    ;*         R0           PATTERN COUNT
5588                                    ;*
5589                                    ;*CALL   JSR     PC,LDDATA
5590                                    ;*      RETURN
5591                                    ;*
5592                                    ;*
5593                                    ;*      THE DATA WILL LOOK AS FOLLOWS:
5594                                    ;*
5595                                    ;*      *----------------------*----------*----------*
5596                                    ;*      !  CPU ID              !  PAT     !  PAT     !  WORD 0
5597                                    ;*      ! NO. OF PAT. WORDS (N) ! NO. OF PAT. WORDS (N) !  WORD 1
5598                                    ;*      !                RANDOM DATA                !  WORDS 2
5599                                    ;*      !                                           !   TO N+1
5600                                    ;*      *-------------------------------------------*
5601                                    ;*      !                 ZERO FILL                 !  WORDS N+2
5602                                    ;*      !                                           !   TO 255
5603                                    ;*      *-------------------------------------------*
5604                                    ;*
5605                                    ;;*****************************************************************
5606
5607  023056  010446                    LDDATA: MOV     R4,-(SP)        ;STORE R4 ON STACK
5608  023060  010346                            MOV     R3,-(SP)        ;STORE R3 ON STACK
5609  023062  010246                            MOV     R2,-(SP)        ;STORE R2 ON STACK
5610  023064  010146                            MOV     R1,-(SP)        ;STORE R1 ON STACK
5611  023066  010046                            MOV     R0,-(SP)        ;STORE R0 ON STACK
5612  023070  016504  000132                    MOV     P.RBAL(R5),R4   ;GET BUS ADDRESS
5613  023074  016501  000130                    MOV     P.RWC(R5),R1    ;STORE WORD COUNT
5614  023100  005401                            NEG     R1              ;GET POSITIVE NUMBER FOR WORD COUNT
5615  023102  122765  000121  000001            CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
5616  023110  001005                            BNE     5$              ;NO, LOAD BUFFER
5617  023112  005024                    3$:     CLR     (R4)+           ;CLEAR BUFFER BEFORE READ
5618  023114  005301                            DEC     R1              ;CHECK IF DONE
5619  023116  001375                            BNE     3$              ;NO, CONTINUE
5620  023120  000137  023270                    JMP     25$             ;RESTORE REGISTERS
5621
5622  023124  116500  000121            5$:     MOVB    P.DPAT(R5),R0   ;STORE PATTERN COUNT
5623  023130  010046                            MOV     R0,-(SP)        ;STORE PATTERN FOR FIRST WORD
5624  023132  006316                            ASL     (SP)            ;MULTIPLY PATTERN BY 4 FOR
5625  023134  006316                            ASL     (SP)            ;  REDUNACY
5626  023136  006316                            ASL     (SP)
5627  023140  006316                            ASL     (SP)
5628  023142  050016                            BIS     R0,(SP)         ;STORE FIRST WORD OF SECTOR
```

```
5629  023144  113766  001064  000001          MOVB    CPUID,1(SP)     ;  ON STACK
5630  023152  006300                          ASL     R0              ;MULTIPLY DATA PATTERN BY 2
5631  023154  016003  004602                  MOV     PATTAB(R0),R3   ;LOAD DATA PATTERN ADDRESS
5632  023160  012700  000040                  MOV     #32.,R0         ;LOAD PATTERN COUNT
5633  023164  011624                  11$:    MOV     (SP),(R4)+      ;STORE PATTERN AND CPU ID IN 1ST WORD
5634  023166  162701  000002                  SUB     #2,R1           ;DECREMENT WORD COUNT BY 2
5635  023172  022701  000376                  CMP     #254.,R1        ;CHECK IF REST OF SECTOR WILL BE
5636                                                                  ;  WRITTEN
5637  023176  101003                          BHI     12$             ;NO, USE R1 AS WORD COUNT
5638  023200  012746  000376                  MOV     #254.,-(SP)     ;STORE SECTOR COUNT = 254
5639  023204  000401                          BR      13$
5640
5641  023206  010146                  12$:    MOV     R1,-(SP)        ;STORE REMAINING WORD COUNT
5642  023210  111666  000001          13$:    MOVB    (SP),1(SP)      ;ADD REDUNDACY
5643  023214  012624                          MOV     (SP)+,(R4)+     ;STORE WORD COUNT IN 2ND WORD
5644  023216  012702  000376                  MOV     #254.,R2        ;LOAD SECTOR WORD COUNT
5645
5646  023222  012324                  15$:    MOV     (R3)+,(R4)+     ;STORE DATA PATTERN
5647  023224  005301                          DEC     R1              ;DECREMENT WORD COUNT
5648  023226  001417                          BEQ     20$             ;BRANCH IF FINISHED
5649  023230  005302                          DEC     R2              ;DECREMENT SECTOR COUNT
5650  023232  005300                          DEC     R0              ;DECREMENT PATTERN COUNT
5651  023234  001407                          BEQ     17$             ;IF ZERO, INITIALIZE PATTERN
5652  023236  005702                          TST     R2              ;ARE WE AT END OF SECTOR
5653  023240  001370                          BNE     15$             ;NO, CONTINUE TO LOAD DATA PATTERN
5654  023242  162703  000074                  SUB     #60.,R3         ;INITIALIZE PATTERN ADDRESS
5655  023246  012700  000040                  MOV     #32.,R0         ;INITIALIZE PATTERN COUNT
5656  023252  000744                          BR      11$             ;START NEXT SECTOR
5657
5658  023254  162703  000100          17$:    SUB     #64.,R3         ;INITIALIZE PATTERN ADDRESS
5659  023260  012700  000040                  MOV     #32.,R0         ;INITIALIZE PATTERN
5660  023264  000756                          BR      15$             ;REPEAT PATTERN
5661
5662  023266  005726                  20$:    TST     (SP)+           ;ADJUST STACK
5663  023270  012600                  25$:    MOV     (SP)+,R0        ;RESTORE R0
5664  023272  012601                          MOV     (SP)+,R1        ;RESTORE R1
5665  023274  012602                          MOV     (SP)+,R2        ;RESTORE R2
5666  023276  012603                          MOV     (SP)+,R3        ;RESTORE R3
5667  023300  012604                          MOV     (SP)+,R4        ;RESTORE R4
5668  023302  000207                          RTS     PC              ;RETURN
```

```
5669                                    .SBTTL  FIND FIRST DATA ERROR
5670
5671  023304  032777  000002  155626  BUFER1: BIT     #SW1,@SWR       ;CHECK IF INHIBIT SOFTWARE DATA
5672                                                                  ;  COMPARE
5673  023312  001017                          BNE     5$              ;YES, RETURN
5674  023314  013765  001362  000220          MOV     SOFCMP,P.CMLG(R5) ;LOAD COMPARISON LENGTH
5675  023322  001413                          BEQ     5$              ;IF NO MORE COMPARISONS, RETURN
5676  023324  016565  000010  000214          MOV     P.BALO(R5),P.ERHD(R5) ;LOAD ADDRESS OF START OF SECTOR
5677  023332  016565  000012  000222          MOV     P.WC(R5),P.BFCP(R5) ;LOAD TRANSFER LENGTH
5678  023340  005465  000222                  NEG     P.BFCP(R5)      ;MAKE IT POSITIVE IT POSITIVE
5679  023344  004737  023362                  JSR     PC,CMPBUF       ;DO COMPARISON
5680  023350  023356                          10$                     ;ERROR RETURN
5681
5682  023352  005720                  5$:     TST     (R0)+           ;ADJUST R0 FOR RETURN
5683  023354  000200                          RTS     R0              ;RETURN
5684
5685  023356  011000                  10$:    MOV     (R0),R0         ;LOAD ERROR RETURN
5686  023360  000200                          RTS     R0              ;RETURN
```

```
5687                                    .SBTTL  COMPARE DATA BUFFER
5688
5689                                    ;;*****************************************************************
5690                                    ;*
5691                                    ;*      THE REGISTERS WILL BE USED AS FOLLOWS:
5692                                    ;*
5693                                    ;*      REGISTER                USE
5694                                    ;*      --------                ---
5695                                    ;*
5696                                    ;*         R5           ADDRESS OF PARAMETER BLOCK
5697                                    ;*         R4           BUFFER ADDRESS
5698                                    ;*         R3           PATTERN ADDRESS
5699                                    ;*         R2           SECTOR WORD COUNT
5700                                    ;*         R1           WORDS ON SECTOR WRITTEN
5701                                    ;*         R0           PATTERN COUNT
5702                                    ;*
5703                                    ;*CALL   JSR     PC,CMPBUF
5704                                    ;*      <ADDRESS OF COMPARISON ERROR RETURN>
5705                                    ;*      RETURN
5706                                    ;*
5707                                    ;*      (SP)    NUMBER OF SOFT DATA COMPARES
5708                                    ;*      (SP+2)  WORD COUNT
5709                                    ;*
5710                                    ;;*****************************************************************
5711
5712  023362  010446                    CMPBUF: MOV     R4,-(SP)        ;STORE R4
5713  023364  010346                            MOV     R3,-(SP)        ;STORE R3
5714  023366  010246                            MOV     R2,-(SP)        ;STORE R2
5715  023370  010146                            MOV     R1,-(SP)        ;STORE R1
5716  023372  010046                            MOV     R0,-(SP)        ;STORE R0
5717  023374  016504  000214                    MOV     P.ERHD(R5),R4   ;GET START ADDRESS
5718  023400  016546  000222                    MOV     P.BFCP(R5),-(SP) ;GET COMPARE COUNT
5719  023404  016546  000220                    MOV     P.CMLG(R5),-(SP) ;LOAD NUMBER OF COMPARISON
5720  023410  001002                            BNE     10$             ;COMPARE IF COMPARE COUNT NOT ZERO
5721  023412  000137  023646                    JMP     40$             ;ELSE RETURN
5722
5723  023416  010465  000214            10$:    MOV     R4,P.ERHD(R5)   ;LOAD SECTOR HEAD IN CASE OF ERROR
5724  023422  005002                            CLR     R2              ;CLEAR OFFSET FROM HEAD
5725  023424  012400                            MOV     (R4)+,R0        ;STORE 1ST WORD (PATTERN INDEX)
5726  023426  010003                            MOV     R0,R3
5727  023430  042700  177760                    BIC     #177760,R0      ;KEEP LOW 4 BITS
5728  023434  042703  177417                    BIC     #177417,R3      ;MASK CPU ID
5729  023440  006203                            ASR     R3              ;SHIFT 4 BITS RIGHT
5730  023442  006203                            ASR     R3
5731  023444  006203                            ASR     R3
5732  023446  006203                            ASR     R3
5733  023450  020003                            CMP     R0,R3           ;CHECK FOR PATTERN MATCH
5734  023452  001066                            BNE     30$             ;NO, REPORT ERROR
5735  023454  005202                            INC     R2              ;INCREMENT OFFSET
5736  023456  005366  000002                    DEC     2(SP)           ;DECREMENT COMPARISON COUNT
5737
5738  023462  006300            13$:    ASL     R0              ;MULTIPLY INDEX BY 2
5739  023464  016003  004602                    MOV     PATTAB(R0),R3   ;STORE PATTERN ADDRESS
5740  023470  126414  000001                    CMPB    1(R4),(R4)      ;CHECK WORD COUNT
5741  023474  001055                            BNE     30$             ;NO, REPORT ERROR
5742  023476  012401                            MOV     (R4)+,R1        ;STORE WORDS IN SECTOR WRITTEN
```

```
5743  023500  042701  177400            BIC     #177400,R1      ;KEEP LOW BYTE
5744  023504  005202                    INC     R2              ;INCREMENT OFFSET VALUE
5745  023506  005366  000002            DEC     2(SP)           ;DECREMENT WORD COUNT
5746  023512  005765  000116            TST     P.SMPL(R5)      ;CHECK IF SAMPLED COMPARE
5747  023516  001467                    BEQ     50$             ;YES, START SAMPLED COMPARE SEQUENCE
5748  023520  005701                    TST     R1              ;CHECK IF SECTOR ALL ZEROES
5749  023522  001426                    BEQ     20$             ;YES GO TO ZERO FILL LOOP
5750  023524  012700  000040            MOV     #32.,R0         ;LOAD PATTERN COUNT
5751  023530  022324            15$:    CMP     (R3)+,(R4)+     ;CHECK PATTERN
5752  023532  001036                    BNE     30$             ;MISCOMPARE ERROR
5753  023534  005366  000002            DEC     2(SP)           ;DECREMENT WORD COUNT
5754  023540  001442                    BEQ     40$             ;NO MORE COMPARISONS
5755  023542  005316                    DEC     (SP)            ;DECREMENT NUMBER OF SOFT
5756                                                            ;  DATA COMPARE
5757  023544  001440                    BEQ     40$             ;NO MORE COMPARISONS
5758  023546  005202                    INC     R2              ;INCREMENT OFFSET
5759  023550  022702  000400            CMP     #256.,R2        ;CHECK IF COMPLETE SECTOR COMPARES
5760  023554  001720                    BEQ     10$             ;YES, START NEXT SECTOR
5761  023556  005301                    DEC     R1              ;DECREMENT NUMBER OF WORDS WRITTEN
5762  023560  001407                    BEQ     20$             ;IF ZERO, GO TO ZERO FILL LOOP
5763  023562  005300                    DEC     R0              ;DECREMENT PATTERN COUNT
5764  023564  001361                    BNE     15$             ;NEXT WORD
5765  023566  012700  000040            MOV     #32.,R0         ;REINITIALIZE PATTERN COUNT
5766  023572  162703  000100            SUB     #64.,R3         ;REINITIALIZE PATTERN ADDRESS
5767  023576  000754                    BR      15$             ;CHECK NEXT WORD
5768
5769  023600  005724            20$:    TST     (R4)+           ;CHECK ZERO FILL OF PATTERN
5770  023602  001012                    BNE     30$             ;MISCOMPARE ERROR
5771  023604  005366  000002            DEC     2(SP)           ;DECREMENT WORD COUNT
5772  023610  001416                    BEQ     40$             ;NO MORE COMPARISONS
5773  023612  005316                    DEC     (SP)            ;DECREMENT NUMBER OF SOFT
5774                                                            ;  DATA COMPARES
5775  023614  001414                    BEQ     40$             ;NO MORE COMPARISONS
5776  023616  005202                    INC     R2              ;INCREMENT OFFSET VALUE
5777  023620  022702  000400            CMP     #256.,R2        ;CHECK IF COMPLETE SECTOR COMPARED
5778  023624  001674                    BEQ     10$             ;YES, START NEXT SECTOR
5779  023626  000764                    BR      20$             ;COMPARE NEXT WORD ON
5780
5781  023630  110265  000217    30$:    MOVB    R2,P.ERAD(R5)   ;STORE OFFSET ADDRESS
5782  023634  022626                    CMP     (SP)+,(SP)+     ;ADJUST STACK
5783  023636  017666  000012  000012    MOV     @12(SP),12(SP)  ;LOAD ERROR RETURN
5784  023644  000404                    BR      45$             ;RETURN
5785
5786  023646  022626            40$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
5787  023650  062766  000002  000012    ADD     #2,12(SP)       ;ADJUST RETURN
5788
5789  023656  105065  000117    45$:    CLRB    P.ECMP(R5)      ;CLEAR ECC COMPARE FLAG
5790  023662  012600                    MOV     (SP)+,R0        ;RESTORE R0
5791  023664  012601                    MOV     (SP)+,R1        ;RESTORE R1
5792  023666  012602                    MOV     (SP)+,R2        ;RESTORE R2
5793  023670  012603                    MOV     (SP)+,R3        ;RESTORE R3
5794  023672  012604                    MOV     (SP)+,R4        ;RESTORE R4
5795  023674  000207                    RTS     PC              ;RETURN
5796
5797  023676  005701            50$:    TST     R1              ;CHECK IF ALL ZEROES
5798  023700  001432                    BEQ     52$             ;YES, TEST FOR ZEROES
```

```
5799  023702  021314                          CMP     (R3),(R4)       ;CHECK FIRST WORD OF PATERN
5800  023704  001351                          BNE     30$             ;MISCOMPARE, REPORT ERROR
5801  023706  026601  000002                  CMP     2(SP),R1        ;CHECK IF ENOUGH WORDS READ FOR
5802                                                                    ;   BOUNDARY COMPARISON
5803  023712  103443                          BLO     55$             ;NO, DO SAMPLE COMPARE
5804  023714  010146                          MOV     R1,-(SP)        ;SAVE SECTOR COUNT
5805  023716  005301                          DEC     R1              ;CALCULATE NUMBER OF WORDS FROM START
5806  023720  006301                          ASL     R1              ;CONVERT TO BYTES
5807  023722  060104                          ADD     R1,R4           ;CALCULATE MEMORY ADDRESS
5808  023724  042701  177700                  BIC     #177700,R1      ;GET PATTERN ADDRESS OFFSET
5809  023730  060103                          ADD     R1,R3           ;CALCULATE PATTERN ADDRESS
5810  023732  021324                          CMP     (R3),(R4)+      ;CHECK IF DATA WORD IS CORRECT
5811  023734  001403                          BEQ     51$             ;YES, CHECK ZERO FILL
5812  023736  005216                          INC     (SP)            ;CALCULATE OFFSET
5813  023740  012602                          MOV     (SP)+,R2        ;STORE OFFSET
5814  023742  000732                          BR      30$             ;REPORT ERROR
5815
5816  023744  012602                  51$:    MOV     (SP)+,R2        ;STORE OFFSET
5817  023746  062702  000002                  ADD     #2,R2           ;CALCULATE OFFSET FOR START OF ZERO FILL
5818  023752  022702  000377                  CMP     #377,R2         ;CHECK IF NO ZERO FILL
5819  023756  103421                          BLO     55$             ;YES, DO SAMPLE COMPARE
5820  023760  026602  000002                  CMP     2(SP),R2        ;CHECK WORD WAS READ
5821  023764  103416                          BLO     55$             ;NO, DO SAMPLE COMPARE
5822  023766  005714                  52$:    TST     (R4)            ;CHECK START OF ZERO FILL
5823  023770  001317                          BNE     30$             ;NO, REPORT ERROR
5824  023772  022766  000377  000002          CMP     #377,2(SP)      ;CHECK IF LAST WORD READ
5825  024000  101010                          BHI     55$             ;NO, DO SAMPLE COMPARE
5826  024002  012702  000377                  MOV     #377,R2         ;LOAD OFFSET COUNT
5827  024006  016504  000214                  MOV     P.ERHD(R5),R4   ;LOAD START OF SECTOR
5828  024012  062704  000776                  ADD     #776,R4         ;CALCULATE END OF SECTOR
5829  024016  005714                          TST     (R4)            ;CHECK LAST WORD OF SECTOR
5830  024020  001303                          BNE     30$             ;REPORT ERROR
5831
5832  024022  016504  000214          55$:    MOV     P.ERHD(R5),R4   ;SET START OF SECTOR
5833  024026  012400                          MOV     (R4)+,R0        ;GET PATTERN INDEX
5834  024030  042700  177760                  BIC     #177760,R0      ;MASK OFF UNNECESSARY BIT
5835  024034  006300                          ASL     R0              ;MULTIPLY BY 2
5836  024036  016003  004602                  MOV     PATTAB(R0),R3   ;GET PATTERN ADDRESS
5837  024042  012401                          MOV     (R4)+,R1        ;GET SECTOR WORD COUNT
5838  024044  042701  177400                  BIC     #177400,R1      ;MASK OFF UNNECESSARY BITS
5839  024050  012702  000002                  MOV     #2,R2           ;INITIALIZE OFFSET
5840  024054  012700  000007                  MOV     #7,R0           ;LOAD NUMBER OF SAMPLES
5841  024060  162701  000043          57$:    SUB     #35.,R1         ;SUBTRACT 35 FROM SECTOR WORD COUNT
5842  024064  101430                          BLOS    60$             ;CHECK IF ZERO FILL
5843  024066  162766  000043  000002          SUB     #35.,2(SP)      ;SUBTRACT 35 FROM WORD COUNT
5844  024074  101664                          BLOS    40$             ;CHECK IF FINISHED
5845  024076  062702  000043                  ADD     #35.,R2         ;CALCULATE OFFSET
5846  024102  062704  000106                  ADD     #106,R4         ;DETERMINE ADDRESS OF COMPARE
5847  024106  062703  000006                  ADD     #6,R3           ;DETERMINE ADDRESS OF PATTERN
5848  024112  021413                          CMP     (R4),(R3)       ;CHECK IF DATA CORRECT
5849  024114  001245                          BNE     30$             ;NO, REPORT ERROR
5850  024116  005316                          DEC     (SP)            ;DECREMENT NUMBER OF COMPARES
5851  024120  001652                          BEQ     40$             ;CHECK IF FINISHED
5852  024122  005300                          DEC     R0              ;CHECK IF IF TIME FOR NEXT SECTOR
5853  024124  001355                          BNE     57$             ;NO, CHECK NEXT SAMPLE
5854  024126  162766  000011  000002          SUB     #11,2(SP)       ;ADJUST WORD COUNT
```

```
5855  024134  101644                        BLOS    40$             ;CHECK IF FINISHED
5856  024136  062704  000022                ADD     #22,R4          ;CALCULATE START OF NEXT SECTOR
5857  024142  000137  023416                JMP     10$             ;GO SERVICE NEXT SECTOR
5858
5859  024146  162766  000043  000002  60$:  SUB     #35.,2(SP)      ;SUBTRACT 35 FROM WORD COUNT
5860  024154  101634                        BLOS    40$             ;CHECK IF FINISHED
5861  024156  062702  000043                ADD     #35.,R2         ;DETERMINE OFFSET
5862  024162  062704  000106                ADD     #106,R4         ;DETERMINE ADDRESS OF COMPARE
5863  024166  005714                        TST     (R4)            ;CHECK IF DATA CORRECT
5864  024170  001217                        BNE     30$             ;NO, REPORT ERROR
5865  024172  005316                        DEC     (SP)            ;DECREMENT NUMBER OF COMPARES
5866  024174  001624                        BEQ     40$             ;BRANCH IF FINISHED
5867  024176  005300                        DEC     R0              ;CHECK IF SAMPLE COUNT EXHAUSTED
5868  024200  001362                        BNE     60$             ;NO, CHECK NEXT SAMPLE
5869  024202  162766  000011  000002        SUB     #11,2(SP)       ;ADJUST WORD COUNT
5870  024210  101616                        BLOS    40$             ;CHECK IF FINISHED
5871  024212  062704  000022                ADD     #22,R4          ;CALCULATE START OF NEXT SECTOR
5872  024216  000137  023416                JMP     10$             ;GO SERVICE NEXT SECTOR
```

```
                                .SBTTL  NORMAL COMMAND RETURN
                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL HANDLE ALL NORMAL COMMAND TERMINATIONS
                                ;*
                                ;*      CALLED ROUTINE          USE
                                ;*      --------------          ---
                                ;*
                                ;*      ASNORM                  ASSIGNMENT NORMAL TERNIMATION
                                ;*      CNTRL1                  CONTROLLER ERROR
                                ;*      ABNORM                  DRIVE ERROR
                                ;*      EVCVY1                  NORMAL ERROR RECOVERY
                                ;*      DATERR                  INCORRECT DATA BUFFER
                                ;*
                                ;*      ASSUMED REGISTERS
                                ;*
                                ;*      REGISTER        CONTENTS
                                ;*      --------        --------
                                ;*
                                ;*      R5              ADDRESS OF PARMETER BLOCK
                                ;*      R2              ADDRESS OF RK06 REGISTERS
                                ;*
                                ;;*****************************************************************
024222  004737  047364          NORMAL: JSR     PC,I.CSTS               ;GATHER CONTROLLER STATUS
024226  032765  024000  000016          BIT     #SPAR!CTO,P.CS1(R5) ;CHECK IF DRIVE BUS PARITY OR CONTROLLER TIME OUT
024234  001010                          BNE     2$                      ;YES, INDICATE CONTROLLER FAULT
                                
                                        CHECK IF ANY OF THE FOLLOWING ERRORS ARE SET IN CS2
                                ;               UNIT FIELD ERROR
                                ;               MULTIPLE DRIVE SELECT
                                ;               PROGRAMMING ERROR
                                ;               NON-EXISTENT MEMORY
                                ;               NON-EXISTENT DRIVE
                                ;               UNIBUS PARITY ERROR
                                ;               WRITE CHECK ERROR
                                ;               DATA LATE
024236  032765  177400  000020          BIT     #UFE!MDS!PGE!NEM!NED!UPE!WCE!DLT,P.CS2(R5)
024244  001004                          BNE     2$                      ;YES, INIDICATE CONTROLLER ERROR
                                
                                        CHECK IF ANY OF THE FOLLOWING ERRORS ARE SET IN RKER
                                ;               ILLEGAL FUNTION CODE
                                ;               FORMAT ERROR
                                ;               DRIVE TYPE ERROR
                                ;               ECC HARD
                                ;               BAD SECTOR ERROR
                                ;               HEADER VRC ERROR
                                ;               CYLINDER ADDRESS OVERFLOW
                                ;               DRIVE TIMING ERROR
                                ;               OPERATION INCOMPLETE
                                ;               DATA CHECK
024246  032765  131761  000034          BIT     #ILC!FMTE!DTYE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,P.ER(R5)
024254  001405                          BEQ     3$                      ;NO, CHECK FOR UNREPORT DRIVE ERRORS
024256  052737  000001  001564  2$:     BIS     #BIT0,ERCONT            ;SET CONTROLLER ERROR NOT FLAGGED
024264  000137  027164                  JMP     CNTRL1                  ;REPORT CONTROLLER ERROR
```

```
                                        ;       CHECK THE FOLLOWING ERROR BITS IN RKER(UNREPORTED DRIVE ERROR)
                                        ;               SEEK INCOMPLETE
                                        ;               NON-EXECUTABLE DRIVE FUNCTION
                                        ;               DRIVE DETECTED DRIVE BUS PARITY ERROR
                                        ;               INVALID DISK ADDRESS
                                        ;               WRITE LOCK ERROR
                                        ;               UNSAFE
5937  024270  032765  046016  000034  3$:   BIT     #SKI!NXF!DRPAR!IDAE!WLE!UNS,P.ER(R5)
5938  024276  001004                        BNE     4$              ;YES,SET UNREPORTED DRIVE ERROR
5939
                                        ;       CHECK THE FOLLOWING ERROR BITS IN RKDS
                                        ;               AC LOW
                                        ;               SPEED LOSS
                                        ;               DRIVE OFF TRACK
5944  024300  032765  000070  000036          BIT     #ACLO!SPDLSS!DROT,P.DS(R5)
5945  024306  001405                        BEQ     5$              ;NO, CHECK IF SPECIAL SEQUENCE
5946  024310  052765  000010  000212  4$:   BIS     #BIT3,P.ERR(R5) ;SET DRIVE ERROR NOT REPORTED
5947  024316  000137  025312                JMP     ABNORM          ;PROCESS ERROR
5948
5949  024322  032765  000001  000014  5$:   BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
5950  024330  001005                        BNE     6$              ;YES, CHECK IF SPECIAL SEQUENCE
5951  024332  052765  000200  000212          BIS     #BIT7,P.ERR(R5) ;SET UNSOLITATED ATTENTION
5952  024340  000137  025312                JMP     ABNORM          ;PROCESS ERROR
5953
5954  024344  005737  001564          6$:   TST     ERCONT          ;CHECK IF CONTROLLER ERROR HAS OCCURRED
5955  024350  100017                        BPL     11$             ;NO, CONTINUE NORMAL PROCESSING
5956  024352  122765  000121  000001        CMPB    #RDDATA,P.CMND(R5) ;CHECK IF DATA TRANSFER COMMAND
5957  024360  101007                        BHI     7$              ;NO, REISSUE COMMAND
5958  024362  132765  000040  000001        BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
5959  024370  001003                        BNE     7$              ;YES, REISSUE COMMAND
5960  024372  004037  025032                JSR     R0,CHKADD       ;CHECK BUS AND DISK ADDRESS
5961  024376  024406                        10$                     ;ERROR RETURN
5962  024400  004037  051406          7$:   JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
5963  024404  001324                        CINITQ                  ;  COMMAND INITIATION QUEUE
5964  024406  000207                10$:   RTS     PC              ;RETURN
5965
5966  024410  005765  000150          11$:  TST     P.DSTT(R5)      ;CHECK IF RETRY SEQUENCE
5967  024414  001417                        BEQ     32$             ;NO, CHECK IF DRIVE IS BEING ASSIGNED
5968  024416  004737  040262                JSR     PC,ERCVY1       ;GO THROUGH ERROR RECOVERY
5969  024422  005737  001566          20$:  TST     ERRPRO          ;CHECK IF ERROR PROCESSING COMPLETE
5970  024426  001367                        BNE     10$             ;NO, RETURN
5971  024430  004037  051466                JSR     R0,Q.POP        ;GET NEXT ERROR TO BE PROCESSED
5972  024434  001330                        ERRPRQ
5973  024436  012605                        MOV     (SP)+,R5        ;LOAD ADDRESS OF PARAMETER BLOCK FOR
5974                                        ;  NEXT ERROR
5975  024440  001762                        BEQ     10$             ;IF NO MORE ERRORS, RETURN
5976  024442  010537  001566                MOV     R5,ERRPRO       ;INDICATE THAT ERROR IS BEING PROCESSED
5977  024446  004737  032002                JSR     PC,ERCVY        ;GO THROUGH ERROR RECOVERY
5978  024452  000763                        BR      20$             ;CHECK IF ERROR PROCESSING COMPLETE
5979
5980  024454  105765  000210          32$:  TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
5981  024460  001402                        BEQ     33$             ;NO, CHECK IF SEEK OR RELEASE
5982  024462  000137  013504                JMP     ASNORM          ;GO CONTINUE ASSIGNMENT
5983
5984  024466  122765  000117  000001  33$:  CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS POSITION
```

```
                                                        ;  (SWITCH 14 OPTION)
024474  001416                          BEQ     34$     ;YES, RELEASE DRIVE
024476  122765  000140  000001          CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
024504  001021                          BNE     35$     ;NO, CHECK ADDRESS AND WORD COUNT
024506  112737  177777  001506          MOVB    #-1,I.ISRL ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
024514  012762  000300  000000          MOV     #INTR,RKCS1(R2) ;FORCE AN INTERRUPT SCAN DRIVE ATTENTIONS
024522  004037  051406                  JSR     R0,G.PUSH ;PUT PARAMTER BLOCK ADDRESS ON
024526  001314                          AVAILQ  ;  DRIVE AVAILIABLE QUEUE
024530  000207                          RTS     PC      ;RETURN

024532  112765  000140  000001  34$:    MOVB    #RELEAS,P.CMND(R5) ;PUT RELEASE COMMAND IN PARAMETER BLOCK
024540  004037  051406                  JSR     R0,Q.PUSH ;ENQUEUE PARAMETER BLOCK IN
024544  001324                          CINITQ  ;  COMMAND INITIATION QUEUE
024546  000207                          RTS     PC      ;RETURN

024550  004037  025032          35$:    JSR     R0,CHKADD ;CHECK WORD COUNT, BUS ADDRESS,
                                                        ;  SECTOR, TRACK, AND CYLINDER
024554  024406                          10$             ;ERROR RETURN ADDRESS
024556  112737  177777  001506          MOVB    #-1,I.ISRL ;INDICATE THAT AN INTERRUPT HAS BEEN ISSUED
024564  012762  000300  000000          MOV     #INTR,RKCS1(R2) ;GO SCAN FOR DRIVE INTERRUPTS
024572  005037  177776                  CLR     PS      ;ALLOW RK06 INTERRUPTS
024576  122765  000121  000001          CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
024604  001004                          BNE     NORM1   ;NO, DO NOT COMPARE BUFFER
024606  004037  023304                  JSR     R0,BUFER1 ;CHECK DATA READ
024612  025424                          DATERR  ;ERROR RETURN
024614  000417                          BR      NORM2   ;RELEASE BUFFER

024616  122765  000131  000001  NORM1:  CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE COMMAND
024624  001013                          BNE     NORM2   ;NO, RELEASE BUFFER
024626  032765  000200  000014          BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
024634  001407                          BEQ     NORM2   ;YES, RELEASE BUFFER
024636  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE FLAG
024644  004037  051406                  JSR     R0,Q.PUSH ;PUT PARAMETER BLOCK IN
024650  001324                          CINITQ  ;  COMMAND INITIATION QUEUE
024652  000207                          RTS     PC      ;RETURN

024654  013737  001452  177776  NORM2:  MOV     RKPRI,PS ;LOCK OUT RK06 INTERRUPTS
024662  004737  022642                  JSR     PC,BUFREL ;RELEASE BUFFER
024666  005037  177776                  CLR     PS      ;ALLOW RK06 INTERRUPTS
024672  032777  040040  154240          BIT     #SW5!SW14,@SWR ;CHECK IF INHIBIT AUTOMATIC DESELECT
024700  001043                          BNE     33$     ;YES, PUT PARAMETER BLOCK IN
                                                        ;  AVAILIABLE QUEUE
024702  026565  000110  000154          CMP     P.MXCD+2(R5),P.NODR+2(R5) ;CHECK IF MAXIMUM NUMBER
                                                        ;  OF COMMAND HAVE BEEN ISSUED
024710  101010                          BHI     29$     ;NO, CHECK IF MAXIMUM NUMBER OF
                                                        ;  OF WORDS HAVE BEEN TRANSFERRED
024712  103404                          BLO     28$     ;YES, DROP DRIVE
024714  026565  000106  000152          CMP     P.MXCD(R5),P.NODR(R5) ;CHECK IF MAXIMUM NUMBER
                                                        ;  OF COMMANDS HAVE BEEN ISSUED
024722  103003                          BHIS    29$     ;NO, CHECK IF MAXIMUM NUMBER OF
                                                        ;  HAVE BEEN TRANSFERRED
024724  004737  013004          28$:    JSR     PC,DROP ;DROP DRIVE
024730  000435                          BR      35$     ;ALLOCATE BUFFER


024732  016546  000166          29$:    MOV     P.NRD+2(R5),-(SP) ;STORE NUMBER OF WORDS READ ON STACK
```

```
 6041  024736  016546  000170                 MOV     P.NRD+4(R5),-(SP)
 6042  024742  066566  000160  000002         ADD     P.NWRT+2(R5),2(SP) ;ADD NUMBER OF WORDS WRITTEN
 6043  024750  005516                         ADC     (SP)
 6044  024752  066516  000162                 ADD     P.NWRT+4(R5),(SP)
 6045  024756  103410                         BCS     30$                ;DROP DRIVE IF OVERFLOW
 6046  024760  026526  000114                 CMP     P.MXWT+2(R5),(SP)+ ;CHECK IF MAXIMUM NUMBER OF WORDS
 6047                                                                     ;  HAVE BEEN TRANSFERRED
 6048  024764  101010                         BHI     32$                ;NO, PUT PARAMETER BLOCK IN
 6049                                                                     ;  AVAILIABLE QUEUE
 6050  024766  103405                         BLO     31$                ;YES, DROP DRIVE
 6051  024770  026526  000112                 CMP     P.MXWT(R5),(SP)+   ;CHECK IF MAXIMUM NUMBER OF WORDS
 6052                                                                     ;  HAVE BEEN TRANSFERRED
 6053  024774  103753                         BLO     28$                ;YES, DROP DRIVE
 6054  024776  000404                         BR      33$                ;NO, PUT PARAMETER BLOCK IN
 6055                                                                     ;  AVAILABLE QUEUE
 6056
 6057  025000  005726                 30$:    TST     (SP)+              ;ADJUST STACK
 6058  025002  005726                 31$:    TST     (SP)+              ;ADJUST STACK
 6059  025004  000747                         BR      28$                ;DROP DRIVE
 6060
 6061  025006  005726                 32$:    TST     (SP)+              ;ADJUST STACK
 6062  025010  112765  000140  000001 33$:    MOVB    #RELEAS,P.CMND(R5) ;PUT RELEASE COMMAND IN PARAMETER BLOCK
 6063  025016  004037  051406                 JSR     R0,Q.PUSH          ;ENQUEUE PARAMETER BLOCK IN
 6064  025022  001324                         CINITQ                     ;  COMMAND INITIATION QUEUE
 6065
 6066  025024  004737  022342         35$:    JSR     PC,GETBUF          ;GET NEW BUFFER
 6067  025030  000207                         RTS     PC                 ;RETURN
```

```
                                        .SBTTL  CHECK BUS ADDRESS, WORD COUNT, SECTOR, TRACK, AND CYLINDER

025032  005765  000022          CHKADD: TST     P.WCR(R5)           ;CHECK IF WORD COUNT ZERO
025036  001404                          BEQ     6$                  ;YES, CHECK BUS ADDRESS
025040  052737  000002  001564          BIS     #BIT1,ERCONT        ;SET WORD COUNT NOT EQUAL ZERO
025046  000512                          BR      24$                 ;REPORT ERROR

025050  032765  001400  000016  6$:     BIT     #BA16!BA17,P.CS1(R5) ;CHECK IF HIGH BUS ADDRESS BITS SET
025056  001011                          BNE     10$                 ;YES, REPORT ERROR
025060  016546  000012                  MOV     P.WC(R5),-(SP)      ;STORE WORD COUNT
025064  005416                          NEG     (SP)                ;MAKE IT POSITIVE
025066  006316                          ASL     (SP)                ;DETERMINE NUMBER OF BITS
025070  066516  000010                  ADD     P.BALO(R5),(SP)     ;DETERMINE EXPECTED BUS ADDRESS
025074  026526  000024                  CMP     P.BAR(R5),(SP)+     ;CHECK BUS ADDRESS
025100  001404                          BEQ     17$                 ;BUS ADDRESS CORRECT--CHECK CYLINDER,
                                                                    ;TRACK, AND SECTOR
025102  052737  000004  001564  10$:     BIS     #BIT2,ERCONT        ;SET BUS ADDRESS INCORRECT
025110  000471                          BR      24$                 ;REPORT ERROR

025112  016546  000012          17$:     MOV     P.WC(R5),-(SP)      ;STORE WORD COUNT
025116  005416                          NEG     (SP)                ;MAKE IT POSITIVE
025120  105716                          TSTB    (SP)                ;CHECK IF EVEN NUMBER OF SECTORS
025122  001402                          BEQ     18$                 ;YES, DO COMPARISON
025124  105266  000001                  INCB    1(SP)               ;INCREMENT SECTOR COUNT
025130  005046                  18$:     CLR     -(SP)               ;MAKE ROOM ON STACK
025132  116616  000003                  MOVB    3(SP),(SP)          ;STORE NUMBER OF SECTORS TRANSFERRED
025136  005066  000002                  CLR     2(SP)               ;CLEAR LOCATION ON STACK
025142  116566  000004  000002          MOVB    P.SECT(R5),2(SP)    ;STORE STARTING SECTOR
025150  066616  000002                  ADD     2(SP),(SP)          ;DETERMINE FINAL SECTOR ADDRESS
025154  005066  000002                  CLR     2(SP)               ;CLEAR NUMBER OF TRACKS TRANSFERRED

025160  022716  000026          19$:     CMP     #22.,(SP)           ;CHECK FOR SECTOR OVERFLOW
025164  101005                          BHI     20$                 ;NO, CHECK IF SECTOR CORRECT
025166  162716  000026                  SUB     #22.,(SP)           ;DECREMENT SECTOR COUNT BY 22 AND
025172  005266  000002                  INC     2(SP)               ;  INCREMENT TRACKS TRANSFERRED
025176  000770                          BR      19$                 ;CHECK FOR SECTOR OVERFLOW

025200  126526  000026          20$:     CMPB    P.DTS(R5),(SP)+     ;CHECK IF FINAL SECTOR CORRECT
025204  001027                          BNE     23$                 ;NO, REPORT ERROR
025206  005046                          CLR     -(SP)               ;MAKE ROOM FOR TRACKS TRANSFERRED
025210  116516  000005                  MOVB    P.TRCK(R5),(SP)     ;STORE STARTING TRACK
025214  066616  000002                  ADD     2(SP),(SP)          ;DETERMINE FINAL TRACK ADDRESS
025220  005066  000002                  CLR     2(SP)               ;CLEAR FINAL CYLINDER

025224  122716  000003          21$:     CMPB    #3,(SP)             ;CHECK FOR TRACK OVERFLOW
025230  101005                          BHI     22$                 ;NO, CHECK FINAL TRACK
025232  162716  000003                  SUB     #3,(SP)             ;DECREMENT TRACK COUNT BY 3 AND
025236  005266  000002                  INC     2(SP)               ;  INCREMENT CYLINDER COUNT
025242  000770                          BR      21$                 ;CHECK FOR TRACK OVERFLOW

025244  126526  000027          22$:     CMPB    P.DTS+1(R5),(SP)+   ;CHECK IF FINAL TRACK CORRECT
025250  001005                          BNE     23$                 ;TRACK INCORRECT ERROR
025252  066516  000002                  ADD     P.CYLN(R5),(SP)     ;CALCULATE FINAL CYLINDER
025256  026516  000030                  CMP     P.DCYL(R5),(SP)     ;CHECK IF CYLINDER CORRECT
025262  001410                          BEQ     25$                 ;CORRECT GO TO NORMAL RETURN
025264  005726                  23$:     TST     (SP)+               ;ADJUST STACK
```

```
 6124  025266  052737  000010  001564          BIS     #BIT3,ERCONT    ;SET INCORRECT SECTOR, TRACK, OR CYLINDER
 6125  025274  004737  027164                  JSR     PC,CNTRL1       ;REPORT ERROR
 6126  025300  011000                          MOV     (R0),R0         ;LOAD R0 FOR ERROR RETURN
 6127  025302  000200                          RTS     R0              ;RETURN
 6128
 6129  025304  005726                  25$:    TST     (SP)+           ;ADJUST STACK
 6130  025306  005720                          TST     (R0)+           ;ADJUST R0
 6131  025310  000200                          RTS     R0              ;NORMAL RETURN
```

```
                                        .SBTTL  ABNORMAL TERMINATION
                                
025312  004737  051626        ABNORM: JSR     PC,Q.RMOV       ;REMOVE PARAMETER BLOCK FROM COMMAND
                                                                ;  INITIATION QUEUE
025316  032765  000100  000014        BIT     #CMDTO,P.PRST(R5) ;CHECK IF COMMAND TIME OUT
025324  001402                        BEQ     3$              ;NO, PROCESS ABNORMAL TERMINATION
025326  000137  025516                JMP     TIMOUT          ;JUMP TO TIME OUT ROUTINE
                                
025332  005737  001564        3$:     TST     ERCONT          ;CHECK IF CONTROLLER ERROR HAS OCCURRED
025336  100010                        BPL     10$             ;NO, CONTINUE ERROR PROCESSING
025340  032765  000001  000014        BIT     #DRVUSE,P.PRST(R5) ;CHECK FOR UNSOLICATED INTERRUPT
025346  001403                        BEQ     5$              ;YES, DO NOT ENQUEUE PARAMETER BLOCK
025350  004037  051406                JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
025354  001324                        CINITQ                  ;  COMMAND INITIATION QUEUE
025356  000207          5$:     RTS     PC                      ;RETURN
                                
025360  032765  000001  000014 10$:     BIT     #DRVUSE,P.PRST(R5) ;CHECK IF DRIVE IN USE
025366  001004                        BNE     ABN000          ;YES, CHECK IF DRIVE BEING ASSIGNED
025370  052765  000200  000212        BIS     #BIT7,P.ERR(R5) ;INDICATE UNSOLICATED INTERRUPT
025376  000420                        BR      ABN001          ;CHECK IF ERROR IS BEING PROCESSED
                                
025400  005765  000150  ABN000: TST     P.DSTT(R5)      ;CHECK IF CURRENTLY UNDERGOING
                                                                ;  ERROR PROCESSING
025404  001415                        BEQ     ABN001          ;NO, CHECK IF ERROR PRESENTLY BEING PROCESSED
025406  100402                        BMI     13$             ;ERROR ENQUEUED
025410  000137  043060                JMP     ERCVY2          ;ERROR RECOVERY ROUTINE
                                
025414  052765  000100  000212 13$:     BIS     #BIT6,P.ERR(R5) ;SET ERROR WHILE ENQUEUED IN ERROR QUEUE
025422  000207                        RTS     PC              ;RETURN
                                
025424  013737  001452  177776 DATERR: MOV     RKPRI,PS        ;LOCK RK06 INTERRUPTS
025432  052765  000004  000212        BIS     #BIT2,P.ERR(R5) ;SET SOFTWARE DETECTED DATA ERROR
                                
025440  012765  100200  000150 ABN001: MOV     #BIT15!BIT7,P.DSTT(R5) ;SET ERROR ENQUEUED AND FIRST ERROR
025446  005737  001566                TST     ERRPRO          ;CHECK IF ERROR IS BEING PROCESSED
025452  001015                        BNE     5$              ;YES, GO ENQUEUE ERROR
025454  010537  001566          2$:   MOV     R5,ERRPRO       ;LOAD ERROR BEING PROCESSED
025460  004737  032002                JSR     PC,ERCVY        ;START ERROR RECOVERY
025464  005737  001566                TST     ERRPRO          ;CHECK IF ERROR PROCESSING THROUGH
025470  001005                        BNE     3$              ;NO, RETURN
025472  004037  051466                JSR     R0,Q.POP        ;GET NEXT ERROR ENQUEUED
025476  001330                        ERRPRQ
025500  012605                        MOV     (SP)+,R5        ;LOAD R5 FOR ERROR PROCESSING
025502  001364                        BNE     2$              ;IF NOT ZERO PROCESS NEXT ERROR
025504  000207          3$:     RTS     PC                      ;RETURN
                                
025506  004037  051406          5$:   JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON
025512  001330                        ERRPRQ                  ;  ERROR PROCESSING QUEUE
025514  000207                        RTS     PC              ;RETURN
```

```
6181                                        .SBTTL  COMMAND TIME OUT ROUTINE
6182
6183  025516  016246  000010          TIMOUT: MOV     RKCS2(R2),-(SP) ;GET CS2 FOR DRIVE NUMBER
6184  025522  042716  177770                  BIC     #^C<DRVMSK>,(SP) ;KEEP DRIVE NUMBER
6185  025526  122665  000000                  CMPB    (SP)+,P.DRVN(R5) ;CHECK IF TIME OUT IF FOR
6186                                                                  ;  CURRENTLY SELECTED DRIVE
6187  025532  001014                          BNE     5$              ;NO, TEST IF LOOPING ON CONTROLLER ERROR
6188  025534  016237  000000  001410          MOV     RKCS1(R2),T.CS1 ;GET COMMAND AND STATUS REGISTER 1
6189  025542  032737  000001  001410          BIT     #GO,T.CS1       ;CHECK IF GO IS STILL SET
6190  025550  001405                          BEQ     5$              ;NO, TEST IF LOOPING ON CONTROLLER ERROR
6191  025552  052737  040000  001462          BIS     #E.CMTO,E.CONT  ;SET CONTROLLER TIMED OUT ON COMMAND
6192  025560  000137  026346                  JMP     CONTRL          ;GO REPORT CONTROLLER ERROR
6193
6194  025564  005737  001564          5$:     TST     ERCONT          ;CHECK IF CYCLING ON CONTROLLER ERROR
6195  025570  100004                          BPL     10$             ;NO, CHECK IF DRIVE IS CURRENTLY
6196                                                                  ;  SEIZED BY OTHER PORT
6197  025572  004037  051406                  JSR     R0,Q.PUSH       ;REQUEUE COMMAND IN COMMAND
6198  025576  001324                          CINITQ                  ;  INITIATION QUEUE
6199  025600  000207                          RTS     PC              ;RETURN
6200
6201  025602  032765  010000  000014  10$:     BIT     #DRVSZD,P.PRST(R5) ;CHECK IF DRIVE WAS SEIZED
6202  025610  001404                          BEQ     12$             ;NO, CONTINUE
6203  025612  052765  000020  000212          BIS     #BIT4,P.ERR(R5) ;SET TIME OUT BECAUSE DRIVE SEIZED
6204  025620  000667                          BR      ABNODO          ;GO INITIATE ERROR PROCESSING
6205
6206  025622  032765  020000  000014  12$:     BIT     #E.UNLD,P.PRST(R5) ;CHECK IF WAITING FOR HEADS TO RELOAD
6207  025630  001404                          BEQ     15$             ;NO, SET TIME OUT FOR POSITIONING
6208  025632  052765  000400  000212          BIS     #BIT8,P.ERR(R5) ;SET TIME OUT IN RELOADING HEADS
6209  025640  000657                          BR      ABNODO          ;GO INITIATE ERROR PROCESSING
6210
6211  025642  052765  000040  000212  15$:     BIS     #BIT5,P.ERR(R5) ;SET TIME OUT DURING POSITIONING
6212  025650  000653                          BR      ABNODO          ;GO INITIATE ERROR PROCESSING
```

```
                                        .SBTTL  PRINT RK06 UNIBUS REGISTERS

                                        PRIREG: MOV     R3,-(SP)        ;STORE R3 ON STACK
                                                MOV     R1,-(SP)        ;STORE R1 ON STACK
                                                MOV     #T.CS1,R3       ;LOAD BEGINNING OF REGISTER STORAGE
                                                MOV     #T.DS,R1        ;LOAD END OF REGISTER STORAGE
                                                TYPE    ,ERR300         ;PRINT REGISTER NAMES
                                                JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
                                                MOV     #T.DB,R1        ;LOAD END OF REGISTER STORAGE
                                                TYPE    ,ERR301         ;PRINT REGISTER NAMES
                                                JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
                                                MOV     (SP)+,R1        ;RESTORE R1
                                                MOV     (SP)+,R3        ;RESTORE R3
                                                RTS     PC              ;RETURN

                                        .SBTTL  PRINT DRIVE STATUS

                                        PRDSTT: MOV     R3,-(SP)        ;STORE R3 ON STACK
                                                TYPE    ,ERR302         ;PRINT REGISTER NAMES
                                                MOV     P.DS(R5),-(SP)  ;CONVERT DRIVE STATUS REG TO OCTAL
                                                JSR     PC,BINOCT
                                                TYPE    ,ERR303         ;PRINT HEADER
                                                MOV     R5,R3           ;DETERMINE START OF PRINT OUT
                                                ADD     #P.ADD,R3
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
                                                JSR     PC,BINOCT
                                                TYPE    ,BLANKS
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
                                                JSR     PC,BINOCT
                                                TYPE    ,ERR304
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
                                                JSR     PC,BINOCT
                                                TYPE    ,BLANKS
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
                                                JSR     PC,BINOCT
                                                TYPE    ,ERR305
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
                                                JSR     PC,BINOCT
                                                TYPE    ,BLANKS
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
                                                JSR     PC,BINOCT
                                                TYPE    ,ERR306
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE A
                                                JSR     PC,BINOCT
                                                TYPE    ,BLANKS
                                                MOV     (R3)+,-(SP)     ;PRINT MESSAGE B
                                                JSR     PC,BINOCT
                                                TYPE    ,$CRLF
                                                MOV     (SP)+,R3        ;RESTORE R3
                                                RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT CONTROLLER STATUS
                                
026074  010346                  PRISTT: MOV     R3,-(SP)        ;STORE R3 ON STACK
026076  010146                          MOV     R1,-(SP)        ;STORE R1 ON STACK
026100  010503                          MOV     R5,R3           ;STORE PARAMETER BLOCK ADDRESS
026102  010501                          MOV     R5,R1           ;  FOR INDEX CACULATIONS
026104  062703  000016                  ADD     #P.CS1,R3       ;CALCULATE BEGINNING OF OUTPUT OCTAL
026110  062701  000036                  ADD     #P.DS,R1        ;CALCULATE END OF OUTPUT OCTAL
026114  104401  064410                  TYPE    ,ERR300         ;PRINT REGISTER NAMES
026120  004737  026132                  JSR     PC,PRSTAT       ;PRINT REGISTER CONTENTS
026124  012601                          MOV     (SP)+,R1        ;RESTORE R1
026126  012603                          MOV     (SP)+,R3        ;RESTORE R3
026130  000207                          RTS     PC              ;RETURN
                                
                                        .SBTTL  PRINT REGISTER CONTENTS
                                
026132  012346                  PRSTAT: MOV     (R3)+,-(SP)     ;SAVE REGISTER VALUE FOR TYPE OUT
026134  004737  052774                  JSR     PC,$OCT         ;CONVERT TO OCTAL
026140  020103                          CMP     R1,R3           ;CHECK IF FINISHED
026142  001403                          BEQ     10$             ;YES, RETURN
026144  104401  060530                  TYPE    ,$BLANKS        ;TYPE 2 BLANKS
026150  000770                          BR      PRSTAT          ;TYPE NEXT REGISTER
                                
026152  104401  001165          10$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
026156  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL  ECC CORRECTION ROUTINE
                                
026160  010446          ECCCOR: MOV     R4,-(SP)        ;STORE R4 ON STACK
026162  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
026164  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
026166  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
026170  013703  001634          MOV     RTYBA,R3        ;LOAD BEGINNING OF SECTOR ADDRESS
026174  013704  001636          MOV     RTYWC,R4        ;LOAD WORDS READ FROM SECTOR
026200  006304                  ASL     R4              ;MAKE IT BYTES
026202  010346                  MOV     R3,-(SP)        ;STORE ADDRESS OF START OF SECTOR
026204  060416                  ADD     R4,(SP)         ;CALCULATE END OF SECTOR
026206  016501  000060          MOV     P.EPOS(R5),R1   ;SAVE POSITION INFORMATION
026212  005301                  DEC     R1              ;DECREMENT BIT COUNT BY ONE TO GET
                                                        ;  BIT ADDRESS
026214  010100                  MOV     R1,R0
026216  006201                  ASR     R1              ;DETERMINE WORDS THAT ARE BAD
026220  006201                  ASR     R1              ;  FOR DATA READ (SHIFT 3 BITS
026222  006201                  ASR     R1              ;  RIGHT)
026224  042701  000001          BIC     #1,R1           ;CLEAR BYTE INDICATOR
026230  020104                  CMP     R1,R4           ;CHECK IF ERROR WITHIN TRANSFER
026232  103037                  BHIS    10$             ;NO, RETURN
026234  060103                  ADD     R1,R3           ;DETERMINE ADDRESS OF ERROR
026236  016537  000062  001622  MOV     P.EPAT(R5),ECCPAT ;LOAD PATTERN
026244  005037  001624          CLR     ECCPAT+2
026250  042700  177760          BIC     #177760,R0      ;DETERMINE BIT POSITION FOR START OF ECC
                                                        ;  CORRECTION
026254  001406                  BEQ     5$              ;CORRECTION STARTS ON
                                                        ;  WORD BOUNDARY
                                
026256  006337  001622  3$:     ASL     ECCPAT          ;SHIFT PATTERN 1 BIT LEFT
026262  006137  001624          ROL     ECCPAT+2
026266  005300                  DEC     R0              ;DECREMENT COUNT
026270  001372                  BNE     3$              ;CHECK IF IN POSITION
                                
026272  011300          5$:     MOV     (R3),R0         ;CORRECT FIRST WORD WITH EXCLUSIVE
026274  013701  001622          MOV     ECCPAT,R1       ;  OF BITS IN ERROR
026300  043713  001622          BIC     ECCPAT,(R3)
026304  040001                  BIC     R0,R1
026306  050123                  BIS     R1,(R3)+
026310  021603                  CMP     (SP),R3         ;CHECK IF SECOND WORD IN TRANSFER
026312  001407                  BEQ     10$             ;NO, RETURN
026314  011300                  MOV     (R3),R0         ;CORRECT SECOND WORD WITH EXCLUSIVE
026316  013701  001624          MOV     ECCPAT+2,R1     ;  OF BITS IN ERROR
026322  043713  001624          BIC     ECCPAT+2,(R3)
026326  040001                  BIC     R0,R1
026330  050113                  BIS     R1,(R3)
                                
026332  005726          10$:    TST     (SP)+           ;ADJUST STACK
026334  012600                  MOV     (SP)+,R0        ;RESTORE R0
026336  012601                  MOV     (SP)+,R1        ;RESTORE R1
026340  012603                  MOV     (SP)+,R3        ;RESTORE R3
026342  012604                  MOV     (SP)+,R4        ;RESTORE R4
026344  000207                  RTS     PC              ;RETURN
```

```
6341                                            .SBTTL  CONTROLLER ERROR DETECTED BY RK06 DRIVER
6342
6343  026346  010446                    CONTRL: MOV     R4,-(SP)        ;STORE R4 ON STACK
6344  026350  004737  047236                    JSR     PC,I.STOR       ;GET CONTROLLER STATUS
6345  026354  032777  002000  152556            BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
6346  026362  001402                            BEQ     1$              ;NO, CONTINUE
6347  026364  104401  001160                    TYPE    ,$BELL          ;RING BELL
6348  026370  013704  001462            1$:     MOV     E.CONT,R4       ;STORE CONTROLLER INDICATORS
6349  026374  032704  100000                    BIT     #E.MDS,R4       ;CHECK IF DUPLICATE DRIVE SELECT
6350  026400  001403                            BEQ     2$              ;NO, CHECK FOR OTHER ERRORS
6351  026402  104401  061002                    TYPE    ,ERR006         ;TYPE "MULTIPLE DRIVE SELECT"
6352  026406  000501                            BR      10$             ;DISPLAY REGISTERS
6353
6354  026410  032704  000001            2$:     BIT     #E.CCLR,R4      ;CHECK IF "CLEAR CONTROLLER DID NOT
6355                                                                    ;  CLEAR ERROR"
6356  026414  001403                            BEQ     3$              ;NO, CHECK OTHER CONTROLLER ERRORS
6357  026416  104401  060560                    TYPE    ,ERR001
6358  026422  000473                            BR      10$             ;DISPLAY REGISTERS
6359
6360  026424  032704  000002            3$:     BIT     #E.NOAT,R4      ;CHECK IF "NO ATTENTION IN RKASOF"
6361  026430  001403                            BEQ     4$              ;NO, CHECK THE OTHER CONTROLLER ERRORS
6362  026432  104401  060625                    TYPE    ,ERR002
6363  026436  000465                            BR      10$             ;DISPLAY REGISTERS
6364
6365  026440  032704  000010            4$:     BIT     #E.UDAT,R4      ;CHECK IF "UNEXPECTED DATA TYPE ERROR"
6366  026444  001403                            BEQ     5$              ;NO, CHECK OTHER CONTROLLER ERRORS
6367  026446  104401  060654                    TYPE    ,ERR003
6368  026452  000457                            BR      10$             ;DISPLAY REGISTERS
6369
6370  026454  032704  000020            5$:     BIT     #E.CLAT,R4      ;CHECK IF "ATTENTION DID NOT RESET
6371                                                                    ;  WITH DRIVE CLEAR"
6372  026460  001403                            BEQ     6$              ;NO, CHECK OTHER CONTROLLER ERRORS
6373  026462  104401  060707                    TYPE    ,ERR004
6374  026466  000451                            BR      10$             ;DISPLAY REGISTERS
6375
6376  026470  032704  000100            6$:     BIT     #E.ILLD,R4      ;CHECK IF "ILLEGAL DRIVER COMMAND"
6377  026474  001403                            BEQ     7$              ;NO, CHECK IF DATA LATE WHEN UNLOADING HEADER
6378  026476  104401  060753                    TYPE    ,ERR005
6379  026502  000443                            BR      10$             ;DISPLAY REGISTERS
6380
6381  026504  032704  000400            7$:     BIT     #E.DLT,R4       ;CHECK IF "DATA LATE WHEN UNLOADING HEADER"
6382  026510  001403                            BEQ     8$              ;NO, CHECK IF "CONTROLLER ERROR WHILE GATERING
6383                                                                    ;  STATUS"
6384  026512  104401  063611                    TYPE    ,ERR200
6385  026516  000435                            BR      10$             ;DISPLAY REGISTERS
6386
6387  026520  032704  001000            8$:     BIT     #E.CERR,R4      ;CHECK IF "CONTROLLER ERROR DURING
6388                                                                    ;  DRIVE SERVICING"
6389  026524  001425                            BEQ     9$              ;NO, CHECK IF CONTROLLER TIME OUT
6390  026526  032737  000040  001426            BIT     #DTYE,T.ER      ;SEE IF DRIVE TYPE ERROR
6391  026534  001003                            BNE     13$             ;BR IF YES
6392  026536  104401  063651                    TYPE    ,ERR201
6393  026542  000423                            BR      10$             ;DISPLAY REGISTERS
6394  026544  132765  000001  000210  13$:     BITB    #BIT0,P.ASSN(R5)        ;SEE IF IN PROCESS OF ASSIGNING
6395  026552  001003                            BNE     14$             ;BR IF YES
6396  026554  104401  063651                    TYPE    ,ERR201         ;ELSE VALID ERROR
```

```
6397  026560  000414                          BR      10$             ;DISPLAY REGS
6398
6399  026562  004737  027030          14$:    JSR     PC,FLPARM       ;FILL PARAM BLOCK WITH CORRECT RK06-07 DATA
6400  026566  012737  000001  027156          MOV     #1,NEWFLG       ;SET FLAG TO SHOW DTYE ERROR ON ASSIGNING
6401  026574  012604                          MOV     (SP)+,R4        ;RESTORE R4
6402  026576  000427                          BR      12$
6403
6404  026600  032704  040000          9$:     BIT     #E.CMTO,R4      ;CHECK IF CONTROLLER TIME OUT
6405  026604  001414                          BEQ     20$             ;NO, CHECK IF DRIVE DETECTED DRIVE BUS PARITY ERROR
6406  026606  104401  064136                  TYPE    ,ERR207
6407
6408  026612  104401  001165          10$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
6409  026616  004737  043670                  JSR     PC,PRITIM       ;PRINT TIME
6410  026622  004737  025652                  JSR     PC,PRIREG       ;PRINT CONTROLLER REGISTERS
6411  026626  104401  060533                  TYPE    ,ERR000         ;TYPE FATAL ERROR
6412  026632  000000                          HALT                    ;HANG ON FATAL ERROR
6413  026634  000776                          BR      .-2
6414
6415  026636  032704  002000          20$:    BIT     #E.DPAR,R4      ;CHECK IF "DRIVE DETECTED DRIVE BUS PARITY ERROR"
6416  026642  001001                          BNE     30$             ;YES, SET DRPAR IN ERROR REGISTER
6417  026644  000000                          HALT
6418
6419  026646  012604                  30$:    MOV     (SP)+,R4        ;RESTORE R4
6420  026650  052765  000010  000034          BIS     #DRPAR,P.ER(R5) ;SET DRIVE DETECTED DRIVE BUS$OTT> PARITY
6421  026656  010346                  12$:    MOV     R3,-(SP)        ;STORE R3 ON THE STACK
6422  026660  005037  001464                  CLR     @.WAIT          ;CLEAR WAITING FOR COMMAND COMPLETION
6423  026664  116503  000000                  MOVB    P.DRVN(R5),R3   ;GET DRIVE NUMBER
6424  026670  005062  000026                  CLR     RKMR1(R2)       ;CLEAR MAINTENANCE REGISTER 1
6425  026674  010362  000010                  MOV     R3,RKCS2(R2)    ;LOAD DRIVE NUMBER
6426  026700  052762  000040  000010          BIS     #SCLR,RKCS2(R2) ;DO SUBSYS CLR
6427  026706  105762  000000          35$:    TSTB    RKCS1(R2)       ;WAIT FOR READY
6428  026712  001775                          BEQ     35$
6429  026714  004737  047236                  JSR     PC,I.STOR       ;STORE REGISTERS
6430  026720  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
6431  026726  001413                          BEQ     37$             ;NO, CHECK IF ATTENTION CLEARED
6432
6433                          ;               CHECK FOR CONTROLLER TYPE ERROR
6434  026730  032737  137400  001412          BIT     #UFE!MDS!PGE!NEM!NED!UPE!DLT,T.CS2
6435  026736  001004                          BNE     36$
6436  026740  032737  130761  001426          BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!DTE!OPI!DCK,T.ER
6437  026746  001412                          BEQ     40$
6438
6439  026750  104401  063651          36$:    TYPE    ,ERR201         ;TYPE CONTROLLER ERROR DURING DRIVE SERVICING
6440  026754  000716                          BR      10$             ;GO REPORT ERROR
6441
6442  026756  136337  001514  001425  37$:    BITB    INTMSK(R3),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
6443  026764  001403                          BEQ     40$             ;YES, REPORT ERROR
6444  026766  104401  060707                  TYPE    ,ERR004         ;TYPE "DRIVE CLEAR DID NOT RESET ATTENTION"
6445  026772  000707                          BR      10$             ;GO REPORT ERROR
6446
6447  026774  146337  001514  001512  40$:    BICB    INTMSK(R3),W.TIME ;CLEAR TIMING CURRENTLY ON THIS DRIVE
6448  027002  006303                          ASL     R3              ;MULTIPLY BY 2
6449  027004  005063  001544                  CLR     W.DRV(R3)       ;CLEAR WATCH DOG TIME
6450  027010  012603                          MOV     (SP)+,R3        ;RESTORE R3
6451  027012  005737  027156                  TST     NEWFLG          ;SEE IF DTYE ERROR ON ASSIGNING
6452  027016  001402                          BEQ     41$             ;BR IF NO
```

```
 6453  027020  000137  024222                 JMP     NORMAL          ;ELSE GO NORMAL RET
 6454  027024  000137  025312         41$:    JMP     ABNORM          ;GO TO ABNORMAL RETURN
 6455
 6456  027030  132765  000004  000007 FLPARM: BITB    #B.CDT,P.CS1H(R5)       ;SEE IF RK06 PARMETER USED LAST
 6457  027036  001407                 BEQ     1$              ;BR IF YES
 6458  027040  142765  000004  000007         BICB    #B.CDT,P.CS1H(R5)       ;ELSE TRY RK06 PARAM BECAUSE OF DTYE
 6459  027046  012765  000632  000070         MOV     #632,P.MXCL(R5)
 6460  027054  000207                         RTS     PC
 6461
 6462  027056  152765  000004  000007 1$:     BISB    #B.CDT,P.CS1H(R5)       ;TRY RK07 PARAM BECAUSE OF DTYE
 6463  027064  012765  001456  000070         MOV     #1456,P.MXCL(R5)
 6464  027072  000207                         RTS PC
 6465
 6466  027074  132765  000004  000007 GETNUM: BITB    #B.CDT,P.CS1H(R5)       ;SEE IF RK06
 6467  027102  001412                 BEQ      1$             ;BR IF YES
 6468
 6469  027104  012737  001456  027154         MOV     #814.,LSTCYL    ;ELSE LOAD RK07 VALUES
 6470  027112  012737  151010  027160         MOV     #<814.*22.*3.>+<22.*2.>,HOLD
 6471  027120  012737  151011  027162         MOV     #<814.*22.*3.>+<22.*2.>+1,HOLD1
 6472  027126  000207                         RTS     PC
 6473
 6474  027130  012737  000632  027154 1$:     MOV     #410.,LSTCYL    ;LOAD RK06 VALUES
 6475  027136  012737  064740  027160         MOV     #<410.*22.*3.>+<22.*2.>,HOLD
 6476  027144  012737  064741  027162         MOV     #<410.*22.*3.>+<22.*2.>+1,HOLD1
 6477  027152  000207                         RTS     PC
 6478
 6479  027154  000000         LSTCYL: 0
 6480  027156  000000         NEWFLG: 0
 6481  027160  000000         HOLD:   0
 6482  027162  000000         HOLD1:  0
 6483
```

```
                                        .SBTTL  CONTROLLER ERROR DUE TO REIGSTER INCONSISTANCIES

027164  004737  051626         CNTRL1: JSR     PC,Q.RMOV       ;REMOVE PARAMETER BLOCK FROM THE
                                                                ;   COMMAND INITIATION QUEUE
027170  032777  002000  151742 .       BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
027176  001402                         BEQ     1$              ;NO, CONTINUE
027200  104401  001160                 TYPE    ,$BELL          ;RING BELL
027204  032777  020000  151726 1$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
027212  001404                         BEQ     2$              ;NO PRINT ERROR MESSAGE
027214  032777  040000  151716         BIT     #SW14,@SWR      ;CHECK IF CYCLE ON OPERATION
027222  001050                         BNE     10$             ;YES, CHECK IF LOOP ON OPERATION
027224  032737  000001  001564 2$:     BIT     #BIT0,ERCONT    ;CHECK IF "CONTROLLER ERROR NOT FLAGGED"
027232  001402                         BEQ     3$
027234  104401  061311                 TYPE    ,ERR013
027240  032737  000002  001564 3$:     BIT     #BIT1,ERCONT    ;CHECK IF "WORD COUNT NOT 0"
027246  001402                         BEQ     4$
027250  104401  061350                 TYPE    ,ERR014
027254  032737  000004  001564 4$:     BIT     #BIT2,ERCONT    ;CHECK IF "BUS ADDRESS INCORRECT"
027262  001402                         BEQ     5$
027264  104401  061401                 TYPE    ,ERR015
027270  032737  000010  001564 5$:     BIT     #BIT3,ERCONT    ;CHECK IF "CYLINDER,TRACK,SECTOR
                                                                ;   INCORRECT"
027276  001402                         BEQ     7$
027300  104401  061425                 TYPE    ,ERR016
027304  032777  020000  151626 7$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
027312  001014                         BNE     10$             ;YES, CHECK IF HALT ON ERROR
027314  004737  043670                 JSR     PC,PRITIM       ;PRINT TIME
027320  004737  026074                 JSR     PC,PRISTT       ;PRINT RK06 REGISTERS
027324  004737  030244                 JSR     PC,PRI002       ;PRINT CURRENT SUPPLIED COMMAND
027330  032737  000001  001564         BIT     #BIT0,ERCONT    ;CHECK IF CONTROLLER ERROR
                                                                ;   NOT FLAGGED
027336  001402                         BEQ     10$             ;NO, DO NOT PRINT PREVOUS COMMAND
027340  004737  030066                 JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
027344  032777  100000  151566 10$:    BIT     #SW15,@SWR      ;CHECK IF HALT ON ERROR
027352  001010                         BNE     11$             ;YES, HALT
027354  032777  040000  151556         BIT     #SW14,@SWR      ;CHECK IF LOOP OPERATION
027362  001005                         BNE     15$             ;YES, CLEAR ERROR
027364  104401  060533                 TYPE    ,ERR000         ;PRINT "FATAL ERROR"
027370  104401  061030                 TYPE    ,ERR007         ;PRINT "SET SWITCH 14 TO CYCLE ON ERROR"
027374  000000                 11$:    HALT
027376  012737  100000  001564 15$:    MOV     #BIT15,ERCONT   ;SET CONTROLLER ERROR PRESENT
027404  012762  100000  000000         MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
027412  005062  000026                 CLR     RKMR1(R2)       ;CLEAR MAINTANENCE REGISTER 1
027416  112737  000005  001410         MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND INTO TEMPORARY CS1
027424  004037  046774                 JSR     R0,I.ISSU       ;GO ISSUE DRIVE CLEAR
027430  027440                         20$                     ;ERROR RETURN
027432  004037  051406                 JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
027436  001324                         CINITQ                  ;   INITIATION QUEUE
027440  000207                 20$:    RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT WHOLE DATA BUFFER

027442  122765  000121  000123  PRIBUF: CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
027450  001017                          BNE     5$                 ;NO, RETURN
027452  032765  000010  000150          BIT     #BIT3,P.DSTT(R5)   ;CHECK IF TIME TO PRINT BUFFER
027460  001013                          BNE     5$                 ;NO, RETURN
027462  052765  000010  000150          BIS     #BIT3,P.DSTT(R5)   ;SET BUFFER ATTEMPTED TO BE PRINTED
027470  032777  020000  151442          BIT     #SW13,@SWR         ;CHECK IF INHIBIT PRINT OUT
027476  001004                          BNE     5$                 ;YES, RETURN
027500  032777  000010  151432          BIT     #SW3,@SWR          ;CHECK IF PRINT SECTOR BEFORE BEGINNING
                                                                   ;  ERROR RECOVERY
027506  001001                          BNE     10$                ;YES, PRINT SECTOR IN ERROR
027510  000207                  5$:     RTS     PC                 ;RETURN

027512  010146                  10$:    MOV     R1,-(SP)           ;STORE R1 ON THE STACK
027514  010346                          MOV     R3,-(SP)           ;STORE R3 ON THE STACK
027516  010446                          MOV     R4,-(SP)           ;STORE R4 ON THE STACK
027520  104401  064321                  TYPE    ,ERR213            ;TYPE ADDDRESS OF BEGINNING OF
                                                                   ;  SECTOR IN ERROR
027524  013703  001634                  MOV     RTYBA,R3           ;GET ADDRESS OF START OF SCTOR
027530  013704  001636                  MOV     RTYWC,R4           ;GET WORD COUNT
027534  010346                          MOV     R3,-(SP)           ;STORE STARTING ADDRESS FOR PRINT OUT
027536  004737  052774                  JSR     PC,BINOCT          ;CONVERT OF OCTAL
027542  104401  064372                  TYPE    ,ERR214            ;TYPE DATA READ
027546  012701  000010          15$:    MOV     #8.,R1             ;SET 8 COLUMNS ON PAGE
027552  104401  001165                  TYPE    ,$CRLF             ;TYPE <CR><LF>
027556  012346                  20$:    MOV     (R3)+,-(SP)        ;GET NEXT WORD OF SECTOR
027560  004737  052774                  JSR     PC,BINOCT          ;CONVERT TO OCTAL
027564  005304                          DEC     R4                 ;CHECK IF FINISHED
027566  001405                          BEQ     30$                ;YES, RESTORE REGISTERS
027570  005301                          DEC     R1                 ;CHECK IF END OF LINE
027572  001765                          BEQ     15$                ;YES, INITIALIZE COLUMN COUNT
027574  104401  060530                  TYPE    ,BLANKS            ;TYPE 2 BLANKS
027600  000766                          BR      20$                ;GET NEXT WORD

027602  104401  001165          30$:    TYPE    ,$CRLF             ;TYPE <CR><LF>
027606  012604                          MOV     (SP)+,R4           ;RESTORE R4
027610  012603                          MOV     (SP)+,R3           ;RESTORE R3
027612  012601                          MOV     (SP)+,R1           ;RESTORE R1
027614  000207                          RTS     PC                 ;RETURN
```

```
                                        .SBTTL  PRINT DRIVE AND PACK SERIAL NUMBERS
                                
                                PRISER: MOV     R1,-(SP)        ;STORE R1 ON STACK
                                        MOV     R3,-(SP)        ;STORE R3 ON STACK
                                        MOV     R4,-(SP)        ;STORE R4 ON STACK
                                        MOV     #STT100,R1      ;GET ADDRESS OF ASCII STRING
                                        MOV     P.SERL(R5),R3   ;GET DRIVE SERIAL NUMBER
                                        ROL     R3              ;SHIFT DRIVE SERIAL NUMBER 4 BITS LEFT
                                        ROL     R3
                                        ROL     R3
                                        ROL     R3
                                        MOV     #3,R4           ;LOAD NUMBER OF DIGITS OF SERIAL NUMBER
                                2$:     CLRB    (R1)            ;CLEAR NEXT BYTE FOR PRINT OUT
                                        ROL     R3              ;SHIFT IN NEXT 4 BITS
                                        ROLB    (R1)
                                        ROL     R3
                                        ROLB    (R1)
                                        ROL     R3
                                        ROLB    (R1)
                                        ROL     R3
                                        ROLB    (R1)
                                        BISB    #60,(R1)+       ;CHANGE BCD TO DECIMAL
                                        DEC     R4              ;CHECK IF FINISHED
                                        BNE     2$              ;CONVERT NEXT DIGIT
                                        TYPE    ,STT000         ;TYPE "DRIVE SERIAL NUMBER"
                                        MOV     P.PKSR(R5),R3   ;GET LEAST SIGNIFICANT BITS OF
                                                                ;  PACK SERIAL NUMBER
                                        MOV     P.PKSR+2(R5),R4 ;GET MOST SIGNIFICANT BITS OF
                                                                ;  PACK SERIAL NUMBER
                                        MOV     #STT012,R1      ;GET ADDRESS OF ASCII STRING
                                        CLRB    (R1)            ;CLEAR FIRST BYTE OF PRINT OUT
                                        BR      7$              ;START CONVERSION TO OCTAL
                                
                                5$:     CLRB    (R1)            ;CLEAR NEXT BYTE OF PRINT OUT
                                        ROL     R3              ;SHIFT IN NEXT THREE BITS
                                        ROL     R4
                                        ROLB    (R1)
                                7$:     ROL     R3
                                        ROL     R4
                                        ROLB    (R1)
                                        ROL     R3
                                        ROL     R4
                                        ROLB    (R1)
                                        BISB    #60,(R1)+       ;MAKE IT ASCII
                                        CMP     #STT013,R1      ;CHECK IF FINISHED
                                        BHI     5$              ;NO, CONTINUE CONVERSION
                                        TYPE    ,STT011         ;TYPE PACK SERIAL NUMBER
                                        MOV     (SP)+,R4        ;RESTORE R4
                                        MOV     (SP)+,R3        ;RESTORE R3
                                        MOV     (SP)+,R1        ;RESTORE R1
                                        RTS     PC              ;RETURN
```

```
                                        .SBTTL  PRINT BEGINNING OF ERROR HEADER
                                
030000  032777  002000  151132  PRI000: BIT     @#SW10,@SWR     ;CHECK IF BELL ON ERROR
030006  001402                          BEQ     1$              ;NO, CHECK IF INHIBIT PRINT OUT
030010  104401  001160                  TYPE    ,$BELL          ;RING BELL
030014  032777  020000  151116  1$:     BIT     @#SW13,@SWR     ;CHECK IF INHIBIT PRINT OUT
030022  001020                          BNE     20$             ;YES, INCREMENT CONTROLLER ERROR
030024  116537  000000  056357          MOVB    P.DRVN(R5),OPR011 ;STORE DRIVE NUMBER
030032  152737  000060  056357          BISB    #60,OPR011      ;MAKE IT ASCII
030040  104401  056340                  TYPE    ,OPR010         ;TYPE "DRIVE NUMBER N"
030044  132765  000001  000210          BITB    #BIT0,P.ASSN(R5)        ;SEE IF SIZING FOR DRIVE
030052  001004                          BNE     20$             ;BR IF YES, TO RETURN
030054  004737  027616                  JSR     PC,PRISER       ;PRINT DRIVE AND PACK SERIAL NUMBERS
030060  004737  043670                  JSR     PC,PRITIM       ;PRINT TIME
030064  000207          20$:    RTS     PC                      ;RETURN
```

```
6639                                            .SBTTL  PRINT PREVIOUS COMMAND
6640
6641  030066  104401  063462            PRI001: TYPE    ,ERR101                 ;TYPE "PREVIOUS CMND CYL TRK SEC  WRD CT"
6642  030072  122765  000121  000135           CMPB    #RDDATA,P.LCMD(R5) ;CHECK IF READ DATA
6643  030100  001402                           BEQ     5$                      ;YES, DO NOT PRINT PATTERN NUM
6644  030102  104401  063544                   TYPE    ,ERR102                 ;PRINT "PAT"
6645  030106  104401  001165            5$:    TYPE    ,$CRLF                  ;PRINT <CR><LF>
6646  030112  005046                           CLR     -(SP)                   ;MAKE ROOM ON STACK
6647  030114  116516  000135                   MOVB    P.LCMD(R5),(SP) ;GET LAST COMMAND
6648  030120  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6649  030124  104401  060530                   TYPE    ,BLANKS                 ;TYPE 2 BLANKS
6650  030130  016546  000136                   MOV     P.LCYL(R5),-(SP) ;GET LAST CYLINDER USED
6651  030134  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6652  030140  104401  060530                   TYPE    ,BLANKS                 ;TYPE 2 BLANKS
6653  030144  005046                           CLR     -(SP)                   ;MAKE ROOM ON STACK
6654  030146  116516  000141                   MOVB    P.LTRK(R5),(SP) ;GET LAST TRACK USED
6655  030152  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6656  030156  104401  060530                   TYPE    ,BLANKS                 ;TYPE 2 BLANKS
6657  030162  005046                           CLR     -(SP)                   ;MAKE ROOM ON STACK
6658  030164  116516  000140                   MOVB    P.LSEC(R5),(SP) ;GET LAST SECTOR USED
6659  030170  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6660  030174  104401  060530                   TYPE    ,BLANKS                 ;TYPE 2 BLANKS
6661  030200  016546  000142                   MOV     P.LWC(R5),-(SP) ;STORE LAST WORD COUNT
6662  030204  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6663  030210  122765  000121  000135           CMPB    #RDDATA,P.LCMD(R5) ;CHECK IF PAT # VALID
6664  030216  001407                           BEQ     10$                     ;NO, DO NOT TYPE
6665  030220  104401  060530                   TYPE    ,BLANKS                 ;TYPE 2 BLANKS
6666  030224  005046                           CLR     -(SP)                   ;MAKE ROOM ON STACK
6667  030226  116516  000144                   MOVB    P.LDPT(R5),(SP) ;GET LAST DATA PATTERN
6668  030232  004737  052774                   JSR     PC,BINOCT               ;CONVERT TO OCTAL
6669  030236  104401  001165            10$:   TYPE    ,$CRLF                  ;TYPE <CR><LF>
6670  030242  000207                           RTS     PC                      ;RETURN
```

```
6671                                        .SBTTL  PRINT CURRENT GENERATED COMMAND
6672
6673  030244  104401  063351        PRI002: TYPE    ,ERR100             ;PRINT "CURRENT SUPPLIED COMMAND"
6674                                                                    ;  CYL TRK SEC OFFSET
6675                                                                    ;  BUS AD  WRD CT"
6676  030250  122765  000121  000123        CMPB    #PDDATA,P.RCMD(R5) ;CHECK IF PAT # IS TO BE PRINTED
6677  030256  001402                        BEQ     5$                  ;NO, DO NOT PRINT HEADER
6678  030260  104401  063544              TYPE    ,ERR102             ;PRINT "PAT"
6679  030264  104401  001165        5$:   TYPE    ,$CRLF              ;PRINT <CR><LF>
6680  030270  005046                        CLR     -(SP)               ;MAKE ROOM ON STACK
6681  030272  116516  000001              MOVB    P.CMND(R5),(SP)     ;STORE COMMAND FOR PRINT OUT
6682  030276  004737  052774              JSR     PC,BINOCT           ;CONVERT TO OCTAL
6683  030302  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6684  030306  016546  000002              MOV     P.CYLN(R5),-(SP)    ;STORE CYLINDER
6685  030312  004737  052774              JSR     PC,BINOCT           ;CONVERT TO OCTAL
6686  030316  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6687  030322  005046                        CLR     -(SP)               ;MAKE ROOM ON STACK
6688  030324  116516  000005              MOVB    P.TRCK(R5),(SP)     ;STORE TRACK
6689  030330  004737  052774              JSR     PC,BINOCT           ;CONVERT TO OCTAL
6690  030334  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6691  030340  005046                        CLR     -(SP)               ;MAKE ROOM ON STACK
6692  030342  116516  000004              MOVB    P.SECT(R5),(SP)     ;STORE SECTOR
6693  030346  004737  052774              JSR     PC,BINOCT           ;CONVERT TO OCTAL
6694  030352  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6695  030356  005046                        CLR     -(SP)               ;MAKE ROOM ON THE STACK
6696  030360  122765  000021  000147        CMPB    #21,P.RERD(R5)      ;CHECK IF OFFSET
6697  030366  101002                        BHI     10$                 ;NO, INDICATE ZERO OFFSET
6698  030370  116516  000006              MOVB    P.OFST(R5),(SP)     ;STORE OFFSET FOR PRINT OUT
6699  030374  004737  052774        10$:  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6700  030400  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6701  030404  016546  000010              MOV     P.BALO(R5),-(SP)    ;STORE BUS ADDRESS
6702  030410  004737  052774              JSR     PC,BINOCT           ;COVERT TO OCTAL
6703  030414  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6704  030420  016546  000012              MOV     P.WC(R5),-(SP)      ;STORE WORD COUNT
6705  030424  004737  052774              JSR     PC,BINOCT           ;COVERT TO OCTAL
6706  030430  122765  000121  000123        CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF PAT # IS TO BE PRINTED
6707  030436  001407                        BEQ     15$                 ;NO, RETURN
6708  030440  104401  060530              TYPE    ,BLANKS             ;TYPE 2 BLANKS
6709  030444  005046                        CLR     -(SP)               ;MAKE ROOM ON THE STACK
6710  030446  116516  000121              MOVB    P.DPAT(R5),(SP)     ;STORE DATA PATTERN FOR PRINT OUT
6711  030452  004737  052774              JSR     PC,BINOCT           ;CONVERT TO OCTAL
6712  030456  104401  001165        15$:  TYPE    ,$CRLF              ;TYPE <CR><LF>
6713  030462  000207                        RTS     PC                  ;RETURN
```

```
                                .SBTTL  PRINT ERROR MESSAGE AND DROP DRIVE

                                ;;*****************************************************************
                                ;*
                                ;*      EXPECTED PRINT CODES
                                ;*      --------------------
                                ;*
                                ;*      BIT 1   PRINT DRIVE STATUS
                                ;*      BIT 2   PRINT PREVIOUS POSITION
                                ;*      BIT 5   QUESTIONABLE DRIVE STATUS
                                ;*      BIT 6   CURRENT GENERATED COMMAND
                                ;*
                                ;*****************************************************************

030464  010446                  PRI100: MOV     R4,-(SP)        ;STORE R4 ON STACK
030466  012004                          MOV     (R0)+,R4        ;STORE ADDRESS OF PRINT CODE
030470  005204                          INC     R4              ;CALCULATE START OF MESSAGE
030472  010437  030504                  MOV     R4,2$           ;STORE ERROR MESSAGE ADDRESS
030476  004737  030000                  JSR     PC,PRI000       ;PRINT HEADER
030502  104401                          TYPE                    ;TYPE ERROR MESSAGE
030504  000000                  2$:     .WORD   0
030506  132765  000001  000210          BITB    #BIT0,P.ASSN(R5)        ;SEE IF SIZING FOR DRIVE
030514  001045                          BNE     12$             ;BR IF YES
030516  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
030522  032777  020000  150410          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINTOUT
030530  001033                          BNE     10$             ;YES, RETURN
030532  105765  000150                  TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
030536  100402                          BMI     3$              ;YES, GO PRINT CONTROLLER REGISTERS
030540  104401  063721                  TYPE    ,ERR202         ;TYPE ERROR DURING ERROR RECOVERY
030544  004737  026074          3$:     JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
030550  132744  000002                  BITB    #BIT1,-(R4)     ;CHECK IF PRINT DRIVE STATUS
030554  001407                          BEQ     5$              ;NO, CONTINUE
030556  132714  000040                  BITB    #BIT5,(R4)      ;CHECK IF QUESTIONABLE STATUS
030562  001402                          BEQ     4$              ;NO, DO NOT PRINT MESSAGE
030564  104401  064102                  TYPE    ,ERR206         ;PRINT "QUESTIONABLE DRIVE STATUS"
030570  004737  025720          4$:     JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
030574  132714  000100          5$:     BITB    #BIT6,(R4)      ;CHECK IF PRINT CURRENT GENERATED COMMAND
030600  001402                          BEQ     8$              ;NO, DO NOT PRINT CURRENT COMMAND
030602  004737  030244                  JSR     PC,PRI002       ;PRINT CURRENT GENERATED COMMAND
030606  132714  000004          8$:     BITB    #BIT2,(R4)      ;CHECK IF PRINT PREVIOUS COMMAND
030612  001402                          BEQ     10$             ;NO, CHECK IF HALT ON ERROR
030614  004737  030066                  JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
030620  004737  031250          10$:    JSR     PC,HLT000       ;TEST IF HALT ON ERROR
030624  005265  000206                  INC     P.NER(R5)       ;INCREMENT OTHER ERROR COUNT
030630  004737  022642          12$:    JSR     PC,BUFREL       ;RELEASE BUFFER
030634  004737  013004                  JSR     PC,DROP         ;DROP DRIVE FROM TEST SEQUENCE
030640  004737  022342                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
030644  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENTLY BEING PROCESSED
030650  012604                          MOV     (SP)+,R4        ;RESTORE R4
030652  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL  PRINT ERROR MESSAGE

                                ;;*****************************************************************
                                ;*
                                ;*      EXPECTED PRINT CODES
                                ;*      --------------------
                                ;*
                                ;*      BIT 0   DATA TYPE ERROR = 1
                                ;*              NON-DATA TYPE ERROR = 0
                                ;*      BIT 1   PRINT DRIVE STATUS
                                ;*      BIT 2   PRINT PREVIOUS POSITION
                                ;*      BIT 3   PRINT SECTOR IN ERROR
                                ;*      BIT 4   PRINT HEADER
                                ;*      BIT 5   QUESTIONABLE DRIVE STATUS
                                ;*      BIT 6   CURRENT GENERATED COMMAND
                                ;*
                                ;;*****************************************************************

030654  010446                  PRI200: MOV     R4,-(SP)        ;STORE R4 ON STACK
030656  012004                          MOV     (R0)+,R4        ;GET PRINT CODE ERROR FLAGS
030660  132724  000001                  BITB    #BIT0,(R4)+     ;CHECK IF DATA TYPE ERROR
030664  001416                          BEQ     5$              ;NO, NON-DATA TYPE ERROR
030666  032765  000002  000150          BIT     #BIT1,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR
                                                                ;  BEING PROCESSED
030674  001402                          BEQ     1$              ;NO, PRINT MESSAGE
030676  012604                          MOV     (SP)+,R4        ;RESTORE R4
030700  000200                          RTS     R0              ;RETURN

030702  052765  000002  000150  1$:     BIS     #BIT1,P.DSTT(R5) ;SET DATA TRANSFER ERROR OCCURRED
030710  100015                          BPL     10$             ;CHECK IF START OF ERROR RECOVERY
030712  052765  000001  000150          BIS     #BIT0,P.DSTT(R5) ;YES, SET SERVICING DATA ERROR
030720  000411                          BR      10$             ;GO REPORT ERROR

030722  032765  000004  000150  5$:     BIT     #BIT2,P.DSTT(R5) ;CHECK IF NON-DATA TYPE ERROR OCCURRED
030730  001402                          BEQ     6$              ;NO, REPORT ERROR
030732  012604                          MOV     (SP)+,R4        ;RESTORE R4
030734  000200                          RTS     R0              ;RETURN

030736  052765  000004  000150  6$:     BIS     #BIT2,P.DSTT(R5) ;SET NON-DATA ERROR OCCURRED
030744  004737  030000          10$:    JSR     PC,PRI000       ;TYPE ERROR HEADER
030750  032777  020000  150162          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
030756  001115                          BNE     30$             ;YES, INHIBIT PRINT OUT
030760  010437  031000                  MOV     R4,12$          ;STORE ERROR MESSAGE
030764  105765  000150                  TSTB    P.DSTT(R5)      ;CHECK IF ERROR DURING RECOVERY
030770  100402                          BMI     11$             ;NO, PRINT ERROR
030772  104401  063721                  TYPE    ,ERR202         ;PRINT "ERROR DURING ERROR RECOVERY"
030776  104401                  11$:    TYPE                    ;PRINT ERROR MESSAGE
031000  000000                  12$:    .WORD   0
031002  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
031006  004737  026074                  JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
031012  132744  000002                  BITB    #BIT1,-(R4)     ;CHECK IF PRINT DRIVE STATUS
031016  001407                          BEQ     22$             ;NO, CONTINUE
031020  132714  000040                  BITB    #BIT5,(R4)      ;CHECK IF QUESTIONABLE DRIVE STATUS
031024  001402                          BEQ     20$             ;NO, DO NOT PRINT MESSAGE
031026  104401  064102                  TYPE    ,ERR206         ;TYPE "QUESTIONABLE DRIVE STATUS"
031032  004737  025720          20$:    JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
```

```
6820  031036  132714  000100          22$:    BITB    #BIT6,(R4)          ;CHECK IF PRINT CURRENT COMMAND
6821  031042  001402                          BEQ     23$                 ;NO, CONTINUE
6822  031044  004737  030244                  JSR     PC,PRI002           ;PRINT CURRENTLY ISSUED COMMAND
6823  031050  132714  000004          23$:    BITB    #BIT2,(R4)          ;CHECK IF PRINT PREVIOUS COMMAND
6824  031054  001402                          BEQ     25$                 ;NO, CHECK IF PRINT HEADER
6825  031056  004737  030066                  JSR     PC,PRI001           ;PRINT PREVIOUS COMMAND
6826  031062  132714  000001          25$:    BITB    #BIT0,(R4)          ;CHECK IF DATA TYPE ERROR
6827  031066  001422                          BEQ     27$                 ;NO, DO NOT PRINT ECC
6828  031070  032765  100000  000034        BIT     #DCK,P.ER(R5)       ;CHECK IF DATA CHECK
6829  031076  001416                          BEQ     27$                 ;NO, DO NOT PRINT ECC
6830  031100  104401  063150                  TYPE    ,ERR067             ;TYPE ECC PAT, ECC POS
6831  031104  016546  000062                  MOV     P.EPAT(R5),-(SP)    ;GET PATTERN
6832  031110  004737  052774                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6833  031114  104401  060530                  TYPE    ,BLANKS
6834  031120  016546  000060                  MOV     P.EPOS(R5),-(SP)    ;GET POSITION
6835  031124  004737  052774                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6836  031130  104401  001165                  TYPE    ,$CRLF
6837  031134  132724  000020          27$:    BITB    #BIT4,(R4)+         ;CHECK IF PRINT HEADER
6838  031140  001424                          BEQ     30$                 ;NO, CHECK IF HALT ON ERROR
6839  031142  104401  063563                  TYPE    ,ERR111             ;TYPE HEADER DESCRIPTOR
6840  031146  013746  001646                  MOV     HEAD1,-(SP)         ;STORE FIRST WORD OF THE HEADER
6841  031152  004737  052774                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6842  031156  104401  060530                  TYPE    ,BLANKS             ;TYPE 2 BLANKS
6843  031162  013746  001650                  MOV     HEAD2,-(SP)         ;STORE SECOND WORD OF THE HEADER
6844  031166  004737  052774                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6845  031172  104401  060530                  TYPE    ,BLANKS             ;TYPE 2 BLANKS
6846  031176  013746  001652                  MOV     HEAD3,-(SP)         ;STORE THIRD WORD OF THE HEADER
6847  031202  004737  052774                  JSR     PC,BINOCT           ;CONVERT TO OCTAL
6848  031206  104401  001165                  TYPE    ,$CRLF              ;TYPE <CR><LF>
6849  031212  132744  000010          30$:    BITB    #BIT3,-(R4)         ;CHECK IF PRINT WHOLE SECTOR
6850  031216  001402                          BEQ     40$                 ;NO, CHECK IF HALT ON ERROR
6851  031220  004737  027442                  JSR     PC,PRIBUF           ;GO PRINT BUFFER
6852  031224  132714  000001          40$:    BITB    #BIT0,(R4)          ;CHECK IF DATA TYPE ERROR
6853  031230  001403                          BEQ     45$                 ;NO, CHECK IF HALT ON ERROR
6854  031232  042765  100200  000150        BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR
6855                                                                      ;  ENQUEUED
6856  031240  004737  031250          45$:    JSR     PC,HLT000           ;CHECK IF HALT ON ERROR
6857  031244  012604                          MOV     (SP)+,R4            ;RESTORE R4
6858  031246  000200                          RTS     R0                  ;RETURN
```

```
                                        .SBTTL  HALT ON ERROR CHECK

031250  032777  100000  147662  HLTOOO: BIT     #SW15,@SWR      ;CHECK IF HALT ON ERROR
031256  001407                  BEQ     10$             ;NO, RETURN
031260  005737  001464          TST     0.WAIT          ;CHECK IF WAITING FOR COMMAND
                                                        ;  COMPLETION
031264  001403                  BEQ     5$              ;NO, HALT
031266  105762  000000          1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
031272  001775                  BEQ     1$
031274  000000                  5$:     HALT
031276  000207                  10$:    RTS     PC              ;RETURN
```

```
6870                                        .SBTTL  OTHER ERROR STATISTICS GATHERING
6871
6872  031300  105765  000150        NERSTT: TSTB    P.DSTT(R5)          ;CHECK IF FIRST ERROR
6873  031304  100057                        BPL     STT5$               ;NO, CONTINUE ERROR RECOVERY
6874  031306  042765  100200 000150        BIC     #BIT15!BIT7,P.DSTT(R5) ;RESET FIRST ERROR AND ERROR ENQUEUED
6875  031314  005265  000206              INC     P.NER(R5)           ;INCREMENT OTHER ERRORS COUNT
6876  031320  022765  000020 000206        CMP     #T.NER,P.NER(R5)    ;CHECK IF ERROR THRESHOLD EXCEEDED
6877  031326  103426                        BLO     STT2$               ;YES, CHECK IF DROP DRIVE
6878  031330  000445                        BR      STT5$               ;NO, START ERROR RECOVERY SEQUENCE
6879
6880                                        .SBTTL  SEEK INCOMPLETE STATISTICS GATHERING
6881
6882  031332  105765  000150        SKISTT: TSTB    P.DSTT(R5)          ;CHECK IF FIRST ERROR
6883  031336  100042                        BPL     STT5$               ;NO, CONTINUE ERROR RECOVERY
6884  031340  005265  000202              INC     P.NSKI(R5)          ;INCREMENT NUMBER OF SEEK INCOMPLETES
6885  031344  000405                        BR      STT1$               ;GO CHECK IF ERROR THRESHOLD EXCEEDED
6886
6887                                        .SBTTL  OPERATION INCOMPLETE STATISTICS GATHERING
6888
6889  031346  105765  000150        OPISTT: TSTB    P.DSTT(R5)          ;CHECK IF FIRST ERROR
6890  031352  100034                        BPL     STT5$               ;NO, CONTINUE ERROR PROCESSING
6891  031354  005265  000204              INC     P.NOPI(R5)          ;INCREMENT NUMBER OF OPERATION INCOMPLETES
6892  031360  042765  100200 000150  STT1$: BIC     #BIT15!BIT7,P.DSTT(R5) ;RESET FIRST ERROR AND ERROR ENQUEUED
6893  031366  016546  000204              MOV     P.NOPI(R5),-(SP)    ;STORE NUMBER OF OPERATION INCOMPLETES
6894  031372  066516  000202              ADD     P.NSKI(R5),(SP)     ;ADD SEEK INCOMPLETES
6895  031376  026526  000104              CMP     P.SKET(R5),(SP)+    ;CHECK IF ERROR THRESHOLD EXCEEDED
6896  031402  103020                        BHIS    STT5$               ;NO, START ERROR RECOVERY
6897  031404  032777  040020 147526  STT2$: BIT     #SW4!SW14,@SWR      ;CHECK IF INHIBIT DRIVE DESELECTION
6898  031412  001014                        BNE     STT5$               ;YES, START RECOVERY SEQUENCE
6899  031414  104401  064047              TYPE    ,ERR20$             ;PRINT "ERROR THRESHOLD EXCEEDED"
6900  031420  004737  022642              JSR     PC,BUFREL           ;RELEASE BUFFER
6901  031424  004737  013004              JSR     PC,DROP             ;DROP DRIVE
6902  031430  004737  022342              JSR     PC,GETBUF           ;GET NEW COMMAND, DRIVE, AND BUFFER
6903  031434  005037  001566              CLR     ERRPRO              ;CLEAR ERROR CURRENT BEING PROCESSED
6904  031440  011000                        MOV     (R0),R0             ;INDICATE NO RECOVERY SEQUENCE IN PROGRESS
6905  031442  000200                        RTS     R0                  ;RETURN
6906
6907  031444  105765  000120        STT5$:  TSTB    P.IERC(R5)          ;CHECK IF INHIBIT ERROR RECOVERY
6908  031450  001010                        BNE     5$                  ;YES, INDICATE UNRECOVERABLE RETURN
6909  031452  105265  000146              INCB    P.RECT(R5)          ;INCREMENT RETRY COUNT
6910  031456  122765  000010 000146        CMPB    #8.,P.RECT(R5)      ;CHECK IF RETRY FINISHED
6911  031464  103402                        BLO     5$                  ;YES, CLEAN UP AFTER UNSUCCESSFUL RETRY
6912  031466  005720                        TST     (R0)+               ;INDICATE RECOVERY SEQUENCE IN PROGRESS
6913  031470  000200                        RTS     R0                  ;RETURN
6914
6915  031472  004737  031502        5$:     JSR     PC,UNRECY           ;CLEAN UP FOR UNSUCCESSFUL RECOVERY
6916  031476  011000                        MOV     (R0),R0             ;INDICATE NO RECOVERY SEQUENCE IN PROGRESS
6917  031500  000200                        RTS     R0                  ;RETURN
```

```
6918                                    .SBTTL  UNSUCESSFUL RECOVERY CLEANUP
6919
6920  031502  105037  001644            UNRECY: CLRB    RTYSCN          ;CLEAR RETRY SECTOR COUNT
6921  031506  004737  022642                    JSR     PC,BUFREL       ;RELEASE BUFFER
6922  031512  032765  000001  000150            BIT     #BIT0,P.DSTT(R5) ;CHECK IF DATA HARD ERROR
6923  031520  001415                            BEQ     1$              ;NO, CHECK IF DRIVE IS BEING ASSIGNED
6924  031522  005265  000200                    INC     P.HARD(R5)      ;INCREMENT HARD ERROR COUNT
6925  031526  026565  000102  000200            CMP     P.UERT(R5),P.HARD(R5) ;CHECK IF HARD ERROR THRESHOLD EXCEEDED
6926  031534  103007                            BHIS    1$              ;NO, CHECK IF DRIVE IS BEING ASSIGNED
6927  031536  032777  040020  147374            BIT     #SW4!SW14,@SWR  ;INHIBIT DRIVE DESELECTION
6928  031544  001003                            BNE     1$              ;YES, CHECK IF DRIVE IS BEING ASSIGNED
6929  031546  104401  064047                    TYPE    ,ERR205         ;PRINT "ERROR THRESHOLD EXCEEDED
6930  031552  000413                            BR      2$              ;GO DROP DRIVE
6931
6932  031554  105765  000210            1$:     TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
6933  031560  001421                            BEQ     5$              ;NO, CLEAN UP ERROR
6934  031562  132765  000010  000210            BITB    #BIT3,P.ASSN(R5) ;CHECK IF READ PACK SERIAL NUMBER
6935  031570  001404                            BEQ     2$              ;NO, DROP DRIVE
6936  031572  132765  000200  000210            BITB    #BIT7,P.ASSN(R5) ;CHECK IF RECAL FOR READ OF
6937                                                                    ;  PACK SERIAL NUMBER
6938  031600  001441                            BEQ     10$             ;NO, ISSUE RECALIBRATE AND TRY AGAIN
6939  031602  104401  064012            2$:     TYPE    ,ERR204         ;TYPE "ERROR RECOVERY UNSUCCESSFUL"
6940  031606  004737  013004                    JSR     PC,DROP         ;DROP DRIVE FROM TEST SEQUENCE
6941  031612  004737  022342                    JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
6942  031616  005037  001566                    CLR     ERRPRO          ;CLEAR ERROR CURRENT BEING PROCESSED
6943  031622  000207                            RTS     PC              ;RETURN
6944
6945  031624  105765  000120            5$:     TSTB    P.IERC(R5)      ;CHECK IF INHIBIT RETRY
6946  031630  001006                            BNE     6$              ;YES, DO NOT PRINT MESSAGE
6947  031632  032777  020000  147300            BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
6948  031640  001002                            BNE     6$              ;YES, CONTINUE
6949  031642  104401  064012                    TYPE    ,ERR204         ;TYPE "ERROR RECOVERY UNSUCCESSFUL"
6950  031646  010446                    6$:     MOV     R4,-(SP)        ;STORE R4 ON STACK
6951  031650  116504  000000                    MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER
6952  031654  156437  001514  001513            BISB    INTMSK(R4),@.OVER ;INDICATE IMPLIED SEEKS ARE ALLOWED
6953  031662  012604                            MOV     (SP)+,R4        ;RESTORE R4
6954  031664  004037  051406                    JSR     R0,@.PUSH       ;PUT PARAMETER BLOCK IN DRIVE
6955  031670  001314                    AVAILQ                          ;  AVAILIABLE QUEUE
6956  031672  004737  022342                    JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
6957  031676  005037  001566                    CLR     ERRPRO          ;CLEAR ERROR CURRENT BEING PROCESSED
6958  031702  000207                            RTS     PC              ;RETURN
6959
6960  031704  122765  000010  000126    10$:    CMPB    #8.,P.RSEC(R5)  ;CHECK IF ALL 16-BIT MODE SECTORS HAVE BEEN TRIED
6961  031712  101733                            BLOS    2$              ;YES, DROP DRIVE
6962  031714  010446                            MOV     R4,-(SP)        ;STORE R4 ON STACK
6963  031716  116504  000000                    MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
6964  031722  156437  001514  001513            BISB    INTMSK(R4),@.OVER ;INDICATE THAT IMPLIED SEEKS ARE ALLOWED
6965  031730  012604                            MOV     (SP)+,R4        ;RESTORE R4
6966  031732  152765  000200  000210            BISB    #BIT7,P.ASSN(R5) ;INDICATE THAT RECALIBRATE HAS BEEN ISSUED
6967  031740  005065  000146                    CLR     P.RECT(R5)      ;CLEAR ERROR FLAGS
6968  031744  005065  000150                    CLR     P.DSTT(R5)
6969  031750  005065  000212                    CLR     P.ERR(R5)
6970  031754  112765  000113  000001            MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE COMMAND
6971  031762  004037  051406                    JSR     R0,@.PUSH       ;PUT PARAMETER BLOCK IN COMMAND
6972  031766  001324                    CINITQ                          ;  INITIATION QUEUE
6973  031770  004737  022342                    JSR     PC,GETBUF       ;GO ALLOCATE MEMORY BUFFER
```

```
 6974  031774  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR PROCESSING IN PROGRESS
 6975  032000  000207                          RTS     PC              ;RETURN
```

```
                                        .SBTTL  ERROR RECOVERY SEQUENCE
                                ;
032002  032765  000200  000212  ERCVY:  BIT     #BIT7,P.ERR(R5) ;CHECK IF UNSOLICITED INTERRUPT
032010  001441                  BEQ     3$              ;NO, CHECK IF ERROR INFORMATION STILL VALID
032012  032777  002000  147120          BIT     #SW10,@SWR      ;CHECK IF BELL ON ERROR
032020  001402                  BEQ     1$              ;NO, CONTINUE PROCESSING
032022  104401  001160                  TYPE    ,$BELL          ;RING BELL
032026  032777  020000  147104  1$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
032034  001020                  BNE     2$              ;YES, CHECK IF HALT ON ERROR
032036  116537  000000  056357          MOVB    P.DRVN(R5),OPR011 ;LOAD DRIVE NUMBER FOR PRINT OUT
032044  152737  000060  056357          BISB    #60,OPR011      ;MAKE IT ASCII
032052  104401  056340                  TYPE    ,OPR010         ;TYPE "DRIVE N"
032056  004737  043670                  JSR     PC,PRITIM       ;PRINT TIME
032062  104401  061663                  TYPE    ,ERR026         ;TYPE "INT FORM DRV NOT UNDER TEST"
032066  004737  026074                  JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
032072  004737  025720                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
032076  004737  031250          2$:     JSR     PC,HLTOOO       ;CHECK IF HALT ON ERROR
032102  005065  000212                  CLR     P.ERR(R5)       ;CLEAR ERROR FLAGS
032106  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
032112  000207                          RTS     PC              ;RETURN
                                ;
032114  032765  000020  000212  3$:     BIT     #BIT4,P.ERR(R5) ;CHECK IF SERVICING DRIVE SEIZED TIME OUT
032122  001427                  BEQ     5$              ;NO, CONTINUE
032124  004737  030000                  JSR     PC,PRIOOO       ;PRINT HEADER
032130  104401  062753                  TYPE    ,ERR059         ;PRINT "TIME OUT BECAUSE DRIVE SEIZED"
032134  032777  020000  146776          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
032142  001002                  BNE     4$              ;YES, DO NOT PRINT CURRENT GENERATED COMMAND
032144  004737  030244                  JSR     PC,PRI002       ;PRINT CURRENT GENERATED COMMAND
032150  005265  000206          4$:     INC     P.NER(R5)       ;INCREMENT OTHER ERRORS
032154  004737  031250                  JSR     PC,HLTOOO       ;CHECK IF HALT ON ERROR
032160  004737  022642                  JSR     PC,BUFREL       ;RELEASE BUFFER
032164  004737  013004                  JSR     PC,DROP         ;DROP DRIVE
032170  004737  022342                  JSR     PC,GETBUF       ;GET NEW DRIVE, COMMAND, AND BUFFER
032174  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
032200  000207                          RTS     PC              ;RETURN
                                ;
032202  032765  000400  000212  5$:     BIT     #BIT8,P.ERR(R5) ;CHECK IF TIME OUT BECAUSE HEADS DID NOT RELOAD
032210  001404                  BEQ     6$              ;NO, CONTINUE
032212  004037  030464                  JSR     R0,PRI100       ;PRINT TIME OUT BECAUSE HEADS DID NOT RELOAD
032216  063306                          ERR074
032220  000207                          RTS     PC              ;RETURN
                                ;
032222  032765  000040  000212  6$:     BIT     #BIT5,P.ERR(R5) ;CHECK IF TIME OUT DURING DRIVE POSITIONING
032230  001425                  BEQ     7$              ;NO, CONTINUE
032232  042765  100000  000150          BIC     #BIT15,P.DSTT(R5) ;RESET ERROR ENQUEUED
032240  052765  000100  000150          BIS     #BIT6,P.DSTT(R5) ;SET READ STATUS ISSUED
032246  116537  000001  001666          MOVB    P.CMND(R5),POSCMD ;STORE COMMAND IN ERROR
032254  042765  000002  000014          BIC     #DRVPOS,P.PRST(R5) ;RESET DRIVE POSITIONING
032262  004737  036746                  JSR     PC,GETCUR       ;STORE PRESENT STATUS FOR PRINT OUT
032266  112765  000141  000001          MOVB    #RDSTAT,P.CMND(R5) ;GET DRIVE STATUS
032274  004037  051406                  JSR     R0,Q.PUSH       ;ENQUEUE COMMAND IN COMMAND
032300  001324                          CINITQ                  ;  INITIATION QUEUE
032302  000207                          RTS     PC              ;RETURN
                                ;
032304  032765  000100  000212  7$:     BIT     #BIT6,P.ERR(R5) ;CHECK IF ERROR WHILE ENQUEUED
032312  001404                  BEQ     8$              ;NO, CONTINUE
```

```
7032  032314  004037  030464                    JSR     R0,PRI100       ;PRINT HEADER
7033  032320  061543                            ERR022                  ;TYPE "ERROR WHILE WAITING TO
7034                                                                    ;  REPORT ERROR"
7035  032322  000207                            RTS     PC              ;RETURN
7036
7037  032324  032765  004000  000016  8$:     BIT     #CTO,P.CS1(R5) ;CHECK IF CONTROLLER TIME OUT
7038  032332  001404                            BEQ     9$              ;NO, CONTINUE
7039  032334  004037  037322                    JSR     R0,CONERR       ;GO REPORT ERROR
7040  032340  061455                            ERR017
7041  032342  000207                            RTS     PC              ;RETURN
7042
7043  032344  032765  002000  000020  9$:     BIT     #PGE,P.CS2(R5) ;CHECK IF PROGRAMMING ERROR
7044  032352  001404                            BEQ     10$             ;NO, CONTINUE
7045  032354  004037  037322                    JSR     R0,CONERR       ;PRINT HEADER
7046  032360  061511                            ERR020                  ;TYPE "PROG ERROR"
7047  032362  000207                            RTS     PC              ;RETURN
7048
7049  032364  032765  000001  000034  10$:    BIT     #ILC,P.ER(R5)  ;CHECK IF ILLEGAL FUNCTION CODE
7050  032372  001404                            BEQ     11$             ;NO, CONTINUE
7051  032374  004037  037322                    JSR     R0,CONERR       ;PRINT HEADER
7052  032400  061524                            ERR021                  ;TYPE "ILLEGAL FUNCTION CODE"
7053  032402  000207                            RTS     PC              ;RETURN
7054
7055  032404  032765  010000  000014  11$:    BIT     #DRVSZD,P.PRST(R5) ;CHECK FOR DRIVE AVAIL LOST
7056  032412  001404                            BEQ     12$             ;BR IF NOT
7057  032414  004037  030464                    JSR     R0,PRI100       ;PRINT HEADER
7058  032420  062002                            ERR029                  ;TYPE "DRIVE BECAME NOT AVAIL"
7059  032422  000207                            RTS     PC              ;RETURN
7060
7061  032424  032765  010000  000020  12$:    BIT     #NED,P.CS2(R5) ;CHECK FOR NON-EXISTENT DRIVE
7062  032432  001404                            BEQ     15$             ;NO, CONTINUE
7063  032434  004037  030464                    JSR     R0,PRI100       ;PRINT HEADER
7064  032440  061610                            ERR023                  ;TYPE "NON-EXISTENT DRIVE"
7065  032442  000207                            RTS     PC              ;RETURN
7066
7067  032444  032765  000400  000020  15$:    BIT     #UFE,P.CS2(R5) ;CHECK IF UNIT FIELD ERROR
7068  032452  001404                            BEQ     18$             ;NO, CONTINUE
7069  032454  004037  030464                    JSR     R0,PRI100       ;PRINT HEADER
7070  032460  061643                            ERR025                  ;TYPE "UNIT FIELD ERROR"
7071  032462  000207                            RTS     PC              ;RETURN
7072
7073  032464  032765  020000  000016  18$:    BIT     #SPAR,P.CS1(R5) ;CHECK IF CONTROLLER DETECTED BAD PARITY
7074  032472  001004                            BNE     19$             ;YES, PRINT MESSAGE
7075  032474  032765  000010  000034          BIT     #DRPAR,P.ER(R5) ;CHECK IF DRIVE DETECTED BAD PARITY
7076  032502  001437                            BEQ     25$             ;NO, CONTINUE
7077  032504  004037  030654          19$:    JSR     R0,PRI200       ;PRINT HEADER
7078  032510  061627                            ERR024                  ;PRINT "SERCON PARITY"
7079  032512  004037  031300                    JSR     R0,NERSTT       ;TAKE STATISTICS FOR OTHER ERRORS
7080  032516  032764                            60$                     ;ERROR RECOVERY SEQUENCE FINISHED RETURN
7081  032520  032765  020000  000016          BIT     #SPAR,P.CS1(R5) ;CHECK IF CONTROLLER DETECTD SERCON PARITY ERROR
7082  032526  001021                            BNE     22$             ;YES, GO RETRY COMMAND
7083  032530  032765  003400  000150          BIT     #BIT8!BIT9!BIT10,P.DSTT(R5) ;CHECK IF NO POSITIONING
7084  032536  001015                            BNE     22$             ;YES, RETRY COMMAND
7085  032540  132765  000007  000210          BITB    #BIT0!BIT1!BIT2,P.ASSN(R5) ;CHECK IF NO POSITION
7086                                                                    ;  DURING DRIVE ASSIGNMENT
7087  032546  001011                            BNE     22$             ;YES, RETRY COMMAND
```

```
7088  032550  042765  030000  000150          BIC     #BIT12!BIT13,P.DSTT(R5) ;RESET RE-SEEK AND RE-OFFSET
7089  032556  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE DRIVE
7090  032564  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE COMMAND
7091  032572  004037  051406          22$:    JSR     R0,Q.PUSH               ;PUT PARAMETER BLOCK IN COMMAND
7092  032576  001324                          CINITQ                          ;  INITIATION QUEUE
7093  032600  000207                          RTS     PC                      ;RETURN
7094
7095  032602  032765  004000  000020  25$:    BIT     #NEM,P.CS2(R5)  ;CHECK IF NON-EXISTENT MEMORY
7096  032610  001404                          BEQ     30$             ;NO, CONTINUE
7097  032612  004037  037322                  JSR     R0,CONERR       ;PRINT HEADER
7098  032616  062277                          ERR044                  ;PRINT "NON-EXISTENT MEMORY"
7099  032620  000207                          RTS     PC              ;RETURN
7100
7101  032622  032765  020000  000020  30$:    BIT     #UPE,P.CS2(R5)  ;CHECK IF UNIBUS PARITY
7102  032630  001404                          BEQ     35$             ;NO, CONTINUE
7103  032632  004037  037322                  JSR     R0,CONERR       ;PRINT HEADER
7104  032636  062315                          ERR045                  ;PRINT "UNIBUS PARITY"
7105  032640  000207                          RTS     PC              ;RETURN
7106
7107  032642  004037  037512          35$:    JSR     R0,TSTDRP       ;CHECK IF DROP DRIVE ERROR
7108  032646  032764                          60$                     ;YES, ERROR RETURN
7109  032650  032765  000002  000034          BIT     #SKI,P.ER(R5)   ;CHECK IF SEEK INCOMPLETE
7110  032656  001407                          BEQ     45$             ;NO, CONTINUE
7111  032660  004037  030654                  JSR     R0,PRI200       ;PRINT HEADER
7112  032664  062201                          ERR038                  ;PRINT "SEEK INCOMPLETE"
7113  032666  004037  031332                  JSR     R0,SKISTT       ;TAKE SEEK INCOMPLETE STATISTICS
7114  032672  032764                          60$                     ;ERROR RECOVERY FINISHED RETURN
7115  032674  000412                          BR      48$             ;GO ISSUE RECALIBRATE
7116
7117  032676  032765  000040  000036  45$:    BIT     #DROT,P.DS(R5)  ;CHECK IF DRIVE OFF TRACK
7118  032704  001420                          BEQ     50$             ;NO, CONTINUE
7119  032706  004037  030654                  JSR     R0,PRI200       ;PRINT HEADER
7120  032712  062216                          ERR039                  ;PRINT "DRIVE OFF TRACK"
7121  032714  004037  031300                  JSR     R0,NERSTT       ;TAKE STATISTICS ON OTHER ERRORS
7122  032720  032764                          60$                     ;END OF ERROR RECOVERY RETURN
7123  032722  042765  034000  000150  48$:    BIC     #BIT11!BIT12!BIT13,P.DSTT(R5) ;RESET STATUS
7124  032730  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
7125  032736  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;LOAD RECALIBRATE
7126  032744  000712                          BR      22$             ;ENQUEUE COMMAND
7127
7128  032746  032765  001000  000034  50$:    BIT     #COE,P.ER(R5)   ;CHECK IF CYLINDER ADDRESS OVERFLOW
7129  032754  001404                          BEQ     60$             ;NO, CONTINUE
7130  032756  004037  037322                  JSR     R0,CONERR       ;PRINTER HEADER
7131  032762  062415                          ERR051                  ;PRINT "CYLINDER ADDRESS OVERFLOW"
7132  032764  000207                  60$:    RTS     PC              ;RETURN
7133
7134  032766  005765  000022          65$:    TST     P.WCR(R5)       ;CHECK IF FINAL WORD COUNT IS ZERO
7135  032772  001410                          BEQ     5$              ;YES, CONTINUE
7136  032774  026565  000022  000012          CMP     P.WCR(R5),P.WC(R5) ;CHECK THAT FINAL WORD COUNT IS LESS
7137                                                                  ;  THAN OR EQUAL TO INITIAL WORD COUNT
7138  033002  103004                          BHIS    5$              ;YES, CONTINUE
7139  033004  004037  037322                  JSR     R0,CONERR       ;TYPE "WORD COUNT INVALID"
7140  033010  063171                          ERR070
7141  033012  000207                          RTS     PC              ;RETURN
7142
7143  033014  032765  001400  000016  5$:     BIT     #BA16!BA17,P.CS1(R5) ;CHECK IF BUS ADDRESS BITS 16 OR 17 SET
```

```
7144  033022  001015                    BNE     8$                  ;YES, REPORT ERROR
7145  033024  026565  000024  000010    CMP     P.BAR(R5),P.BALO(R5) ;CHECK THAT FINAL BUS ADDRESS IS GREATER
7146                                              ;   THAN OR EQUAL TO INITIAL BUS ADDRESS
7147  033032  103411                    BLO     8$                  ;NO, REPORT ERROR
7148  033034  016546  000012            MOV     P.WC(R5),-(SP)      ;PUT INITIAL WORD COUNT ON STACK
7149  033040  005416                    NEG     (SP)                ;MAKE IT POSITIVE
7150  033042  006316                    ASL     (SP)                ;MULTIPLY BY 2
7151  033044  066516  000010            ADD     P.BALO(R5),(SP)     ;CALCULATE MAXIMUM FINAL ADDRESS
7152  033050  026526  000024            CMP     P.BAR(R5),(SP)+     ;MAKE SURE IT IS NOT TO LARGE
7153  033054  101404                    BLOS    10$                 ;GOOD ADDRESS CONTINUE
7154  033056  004037  037322    8$:     JSR     R0,CONERR           ;TYPE "BUS ADDRESS INVALID"
7155  033062  063214                    ERR071
7156  033064  000207                    RTS     PC                  ;RETURN
7157
7158  033066  026565  000030  000002  10$:   CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK THAT FINAL CYLINDER IS GREATER
7159                                              ;   THAN OR EQUAL TO INITIAL CYLINDER
7160  033074  001405                    BEQ     15$                 ;IF EQUAL CHECK TRACK/SECTOR
7161  033076  101014                    BHI     E1$                 ;IF GREATER THAN, DETERMINE ERROR
7162  033100  004037  037322            JSR     R0,CONERR           ;TYPE "CYLIN ADDRESS INVALID"
7163  033104  063240                    ERR072
7164  033106  000207                    RTS     PC                  ;RETURN
7165
7166  033110  026565  000026  000004  15$:   CMP     P.DTS(R5),P.SECT(R5) ;CHECK TRACK/SECTOR GREATER THAN OR
7167                                              ;   EQUAL TO INITIAL VALUES
7168  033116  103004                    BHIS    E1$                 ;YES, GO INTO ERROR RECOVERY
7169  033120  004037  037322            JSR     R0,CONERR           ;TYPE "SECTOR/TRACK INVALID"
7170  033124  063261                    ERR073
7171  033126  000207                    RTS     PC
7172
7173  033130  032765  010000  000034  E1$:    BIT     #DTE,P.ER(R5)       ;CHECK FOR DRIVE TIMING ERROR
7174  033136  001002                    BNE     2$                  ;YES, PROCESS DRIVE TIMING ERROR
7175  033140  000137  033632            JMP     E2$                 ;NO, CONTINUE
7176
7177  033144  105765  000147    2$:     TSTB    P.RERD(R5)          ;CHECK IF FIRST DATA ERROR IN
7178                                              ;   RETRY SEQUENCE
7179  033150  001117                    BNE     30$                 ;NO, REPORT ERROR
7180  033152  016537  000010  001634    MOV     P.BALO(R5),RTYBA    ;STORE SUPPLIED BUS ADDRESS
7181  033160  016537  000012  001636    MOV     P.WC(R5),RTYWC ;STORE SUPPLIED WORD COUNT
7182  033166  005437  001636            NEG     RTYWC               ;MAKE IT POSITIVE
7183  033172  022737  000400  001636    CMP     #256.,RTYWC         ;CHECK IF LESS THAT OR EQUAL TO A SECTOR
7184  033200  103013                    BHIS    21$                 ;YES, WORK WITH ONE SECTOR
7185  033202  026565  000026  000004    CMP     P.DTS(R5),P.SECT(R5) ;CHECK IF STARTING TRACK AND SECTOR
7186  033210  001021                    BNE     25$                 ;NOT RETRY 2 SECTORS
7187  033212  026565  000030  000002    CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK IF STARTING CYLINDER
7188  033220  001015                    BNE     25$                 ;NO, RETRY 2 SECTORS
7189  033222  012737  000400  001636    MOV     #256.,RTYWC         ;SET WORD COUNT = 1 SECTOR
7190  033230  112737  000001  001644  21$:   MOVB    #1,RTYSCN           ;SET SECTOR COUNT = 1
7191  033236  016537  000004  001640    MOV     P.SECT(R5),RTYSEC   ;LOAD SECTOR AND TRACK
7192  033244  016537  000002  001642    MOV     P.CYLN(R5),RTYCYL   ;LOAD CYLINDER
7193  033252  000456                    BR      30$                 ;GO REPORT ERROR
7194
7195  033254  016537  000026  001640  25$:   MOV     P.DTS(R5),RTYSEC    ;SET SECTOR AND TRACK
7196  033262  016537  000030  001642    MOV     P.DCYL(R5),RTYCYL   ;GET CYLINDER
7197  033270  105737  001640            TSTB    RTYSEC              ;CHECK IF SECTOR = 0
7198  033274  001403                    BEQ     26$                 ;YES, GO DECREMENT TRACK
7199  033276  105337  001640            DECB    RTYSEC              ;DECREMENT SECTOR
```

```
7200  033302  000416                          BR      28$             ;DETERMINE WORD COUNT
7201
7202  033304  112737  000025  001640  26$:    MOVB    #21.,RTYSEC     ;LOAD SECTOR = 21
7203  033312  105737  001641                  TSTB    RTYTRK          ;CHECK IF TRACK 0
7204  033316  001403                          BEQ     27$             ;YES, GO DECREMENT CYLINDER
7205  033320  105337  001641                  DECB    RTYTRK          ;DECREMENT TRACK
7206  033324  000405                          BR      28$             ;DETERMINE WORD COUNT
7207
7208  033326  112737  000002  001641  27$:    MOVB    #2,RTYTRK       ;LOAD TRACK = 2
7209  033334  005337  001642                  DEC     RTYCYL          ;DECREMENT CYLINDER
7210  033340  004737  037210          28$:    JSR     PC,CALWC        ;CALCULATE RETRY WORD COUNT
7211  033344  112737  000002  001644          MOVB    #2,RTYSCN       ;LOAD SECTOR COUNT = 2
7212  033352  022737  001000  001636          CMP     #512.,RTYWC     ;CHECK IF LESS THAN OR EQUAL
7213  033360  101004                          BHI     29$             ;  TWO SECTORS
7214  033362  012737  001000  001636          MOV     #512.,RTYWC     ;MAKE IT 512
7215  033370  000407                          BR      30$             ;GO REPORT ERROR
7216
7217  033372  022737  000400  001636  29$:    CMP     #256.,RTYWC     ;CHECK IF ONLY ONE SECTOR
7218  033400  103403                          BLO     30$             ;NO, GO REPORT ERROR
7219  033402  112737  000001  001644          MOVB    #1,RTYSCN       ;LOAD SECTOR COUNT = 1
7220
7221  033410  004037  030654          30$:    JSR     R0,PRI200       ;PRINT HEADER
7222  033414  062330                          ERR046                  ;PRINT DRIVE TIMING ERROR
7223  033416  105765  000120                  TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
7224  033422  001016                          BNE     35$             ;YES, GO CLEAN UP FOR NEXT COMMAND
7225  033424  105265  000147                  INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
7226  033430  122765  000043  000147          CMPB    #43,P.RERD(R5)  ;CHECK IF RETRY UNSUCCESSFUL
7227  033436  101410                          BLOS    35$             ;YES, GO INDICATE UNSUCCESSFUL
7228                                                                  ;  RECOVERY
7229  033440  122765  000121  000123          CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
7230  033446  001407                          BEQ     37$             ;YES, GO TO RECOVERY SEQUENCE
7231  033450  122765  000021  000147          CMPB    #21,P.RERD(R5)  ;CHECK IF RETRY UNSUCCESSFUL
7232  033456  101026                          BHI     40$             ;NO, GO TO RETRY SEQUENCE
7233  033460  004737  031502          35$:    JSR     PC,UNRECY       ;CLEAN UP FOR UNSUCCESSFUL RECOVERY
7234  033464  000207                          RTS     PC              ;RETURN
7235
7236  033466  122765  000021  000147  37$:    CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET +400 MICRO-INCHES
7237  033474  001017                          BNE     40$             ;NO, CONTINUE RECOVERY
7238  033476  122737  000001  001644          CMPB    #1,RTYSCN       ;CHECK IF NUMBER OF SECTORS = 1
7239  033504  001404                          BEQ     38$             ;YES, DO OFFSET
7240  033506  122737  000025  001640          CMPB    #21.,RTYSEC     ;CHECK IF MID-TRANSFER SEEK
7241                                                                  ;  IS REQUIRED
7242  033514  001761                          BEQ     35$             ;YES, INDICATE UNSUCCESSFUL RETRY
7243  033516  010446                  38$:    MOV     R4,-(SP)        ;STORE R4 ON THE STACK
7244  033520  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
7245  033524  146437  001514  001513          BICB    INTMSK(R4),O.OVER ;DO NOT SEEK BEFORE DATA
7246                                                                  ;  TRANSFER (RESETTING OFFSET)
7247  033532  012604                          MOV     (SP)+,R4        ;RESTORE R4
7248
7249  033534  004037  037026          40$:    JSR     R0,DETOFF       ;DETERMINE IF OFFSET INSTRUCTION
7250  033540  033630                          50$                     ;OFFSET ISSUED RETURN
7251  033542  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;LOAD COMMAND
7252  033550  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
7253  033556  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
7254  033564  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
7255  033572  013765  001636  000012          MOV     RTYWC,P.WC(R5)  ;LOAD WORD COUNT
```

```
 7256  033600  005465  000012                  NEG     P.WC(R5)                ;MAKE IT NEGATIVE
 7257  033604  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
 7258  033612  001003                          BNE     45$                     ;NO, ISSUE COMMAND
 7259  033614  052765  000200  000014          BIS     #W.WCK,P.PRST(R5)  ;YES, SET ISSUE WRITE COMMAND
 7260  033622  004037  051406          45$:    JSR     R0,Q.PUSH               ;ENQUEUE PARAMETER BLOCK IN
 7261  033626  001324                                  CINITQ                  ;  COMMAND INITIATION QUEUE
 7262  033630  000207                  50$:    RTS     PC                      ;RETURN
 7263
 7264  033632  032765  020400  000034  E2$:    BIT     #OPI!HVRC,P.ER(R5) ;CHECK IF HEADER TYPE ERROR
 7265  033640  001002                          BNE     2$                      ;YES, PROCESS ERROR
 7266  033642  000137  034170                  JMP     E3$                     ;NO, CONTINUE
 7267
 7268  033646  004737  036746          2$:     JSR     PC,GETCUR               ;GET CURRENT REGISTER STATUS
 7269  033652  112737  000001  001644          MOVB    #1,RTYSCN               ;LOAD RETRY SECTOR COUNT TO 1
 7270  033660  016537  000026  001640          MOV     P.DTS(R5),RTYSEC ;STORE CURRENT SECTOR AND TRACK
 7271  033666  016537  000030  001642          MOV     P.DCYL(R5),RTYCYL ;STORE CURRENT SECTOR
 7272  033674  016537  000010  001634          MOV     P.BALO(R5),RTYBA ;GET INITIAL BUS ADDRESS
 7273  033702  016537  000012  001636          MOV     P.WC(R5),RTYWC  ;GET INITIAL WORD COUNT
 7274  033710  005437  001636                  NEG     RTYWC                   ;MAKE IT POSITIVE
 7275  033714  022737  000400  001636          CMP     #256.,RTYWC             ;CHECK IF LESS THEN OR EQUAL TO ONE SECTOR
 7276  033722  103021                          BHIS    20$                     ;RETRY COMMAND
 7277  033724  026565  000026  000004          CMP     P.DTS(R5),P.SECT(R5) ;CHECK IF CURRENT SECTOR AND TRACK EQUALS
 7278                                                                          ;  INITIAL TRACK AND SECTOR
 7279  033732  001004                          BNE     15$                     ;NO, CALCULATE DESIRED SECTOR
 7280  033734  026565  000030  000002          CMP     P.DCYL(R5),P.CYLN(R5) ;CHECK IF CURRENT CYLINDER EQUALS
 7281                                                                          ;  INITIAL CYLINDER
 7282  033742  001406                          BEQ     18$                     ;YES, USE INITIAL CYLINDER, TRACK, AND SECTOR
 7283  033744  004737  037210          15$:    JSR     PC,CALWC                ;CALCULATE WORD COUNT
 7284  033750  022737  000400  001636          CMP     #256.,RTYWC             ;CHECK IF WORD COUNT LESS THAN OR EQUAL
 7285                                                                          ;  TO ONE SECTOR
 7286  033756  103003                          BHIS    20$                     ;YES, USE CALCULATED WORD COUNT
 7287  033760  012737  000400  001636  18$:    MOV     #256.,RTYWC             ;USE 256 WORDS
 7288  033766  016546  000052          20$:    MOV     P.BIO(R5),-(SP) ;STORE DRIVE CYLINDER ADDRESS
 7289  033772  042716  160017                  BIC     #^C<M.CADD>,(SP) ;THROW AWAY PARITY AND MESSAGE TYPE
 7290  033776  006216                          ASR     (SP)                    ;SHIFT 4 BITS RIGHT
 7291  034000  006216                          ASR     (SP)
 7292  034002  006216                          ASR     (SP)
 7293  034004  006216                          ASR     (SP)
 7294  034006  022637  001642                  CMP     (SP)+,RTYCYL    ;CHECK IF CYLINDER IS CORRECT
 7295  034012  001420                          BEQ     30$                     ;YES, ISSUE READ HEADER
 7296  034014  004037  030654                  JSR     R0,PRI200               ;REPORT ERROR
 7297  034020  062233                          ERR040                  ;TYPE "MISPOS"
 7298  034022  004037  031346                  JSR     R0,OPISTT               ;GET OPI/MISPOSITIONING STATISTICS
 7299  034026  034166                          60$                     ;ERROR THRESHOLD EXCEEDED RETURN
 7300  034030  052765  004000  000150          BIS     #BIT11,P.DSTT(R5)  ;SET RECALIBRATE ISSUED
 7301  034036  112765  000113  000001          MOVB    #RECAL,P.CMND(R5)  ;LOAD RECALIBRATE IN PARAMETER BLOCK
 7302  034044  004037  051406                  JSR     R0,Q.PUSH               ;PUT PARAMETER BLOCK IN COMMAND
 7303  034050  001324                                  CINITQ                  ;  INITIATION QUEUE
 7304  034052  000207                          RTS     PC                      ;RETURN
 7305
 7306  034054  010446                  30$:    MOV     R4,-(SP)                ;STORE R4 ON THE STACK
 7307  034056  113704  001641                  MOVB    RTYTRK,R4               ;GET TRACK FOR INDEX
 7308  034062  136465  001507  000057          BITB    H.HEAD(R4),P.BIT1(R5) ;CHECK IF CORRECT HEAD IS SELECTED
 7309  034070  001013                          BNE     35$                     ;YES, CONTINUE
 7310  034072  012604                          MOV     (SP)+,R4                ;RESTORE R4
 7311  034074  004037  030654                  JSR     R0,PRI200               ;PRINT HEAD SELECT ERROR
```

```
7312  034100  063105                        ERR065
7313  034102  004037  031346                JSR     R0,OPISTT       ;RECORD AS AN OPI
7314  034106  034166                        60$                     ;ERROR THRESHOLD EXCEEDED RETURN
7315  034110  004037  051406                JSR     R0,Q.PUSH       ;REISSUE COMMAND
7316  034114  001324                        CINITQ
7317  034116  000207                        RTS     PC              ;RETURN
7318
7319  034120  012604              35$:      MOV     (SP)+,R4        ;RESTORE R4
7320  034122  052765  000020  000150        BIS     #BIT4,P.DSTT(R5) ;SET READ ALL HEADERS ISSUED
7321  034130  013765  001640  000004        MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
7322  034136  013765  001642  000002        MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
7323  034144  012765  066006  000010        MOV     #HDBUFF,P.BALO(R5) ;LOAD ADDRESS OF HEADER BUFFER
7324  034152  112765  000164  000001        MOVB    #RDALHD,P.CMND(R5) ;LOAD COMMAND
7325  034160  004037  051406                JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK IN COMMAND
7326  034164  001324                        CINITQ                  ;  INITIATION QUEUE
7327  034166  000207              60$:      RTS     PC              ;RETURN
7328
7329  034170  032765  000200  000034  E3$:  BIT     #BSE,P.ER(R5)   ;CHECK FOR BAD SECTOR ERROR
7330  034176  001403                        BEQ     5$              ;NO, CONTINUE
7331  034200  004737  042356                JSR     PC,BDSECT       ;SERVICE BAD SECTOR
7332  034204  000207                        RTS     PC              ;RETURN
7333
7334  034206  032765  100000  000020  5$:   BIT     #DLT,P.CS2(R5)  ;CHECK IF DATA LATE
7335  034214  001444                        BEQ     E4$             ;NO, CONTINUE
7336  034216  004737  030000                JSR     PC,PRIOOO       ;TYPE HEADER
7337  034222  032777  020000  144710        BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
7338  034230  001004                        BNE     10$             ;YES, CHECK IF HALT ON ERROR
7339  034232  104401  062350                TYPE    ,ERR047         ;TYPE "DATA LATE"
7340  034236  004737  026074                JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
7341  034242  004737  031250        10$:    JSR     PC,HLTOOO       ;CHECK IF HALT ON ERROR
7342  034246  005237  001626                INC     DLTCNT          ;INCREMENT DATA LATE COUNT
7343  034252  022737  001000  001626        CMP     #T.DLT,DLTCNT   ;CHECK IF THRESHOLD EXCEEDED
7344  034260  103006                        BHIS    15$             ;NO, REISSUE COMMAND
7345  034262  104401  060533                TYPE    ,ERROOO         ;TYPE FATAL ERROR
7346  034266  104401  061136                TYPE    ,ERR009         ;TYPE DATA LATE THRESHOLD EXCEEDED
7347  034272  000000                        HALT                    ;HANG
7348  034274  000776                        BR      .-2
7349
7350  034276  105765  000150        15$:    TSTB    P.DSTT(R5)      ;CHECK IF FIRST ERROR
7351  034302  100005                        BPL     20$             ;NO, REISSUE COMMAND
7352  034304  042765  100200  000150        BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR ERROR ENQUEUED AND FIRST ERROR
7353  034312  005037  001566                CLR     ERRPRO          ;CLEAR ERROR RECOVERY IN PROGRESS
7354  034316  004037  051406        20$:    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
7355  034322  001324                        CINITQ                  ;  INITIATION QUEUE
7356  034324  000207                        RTS     PC              ;RETURN
7357
7358  034326  032765  100000  000034  E4$:  BIT     #DCK,P.ER(R5)   ;CHECK IF DATA CHECK
7359  034334  001002                        BNE     2$              ;YES, START RECOVERY
7360  034336  000137  035136                JMP     E5$             ;NO, CONTINUE
7361
7362  034342  112737  000001  001644  2$:   MOVB    #1,RTYSCN       ;SET SECTOR COUNT = 1
7363  034350  016537  000004  001640        MOV     P.SECT(R5),RTYSEC ;STORE RETRY SECOTOR AND TRACK
7364  034356  016537  000002  001642        MOV     P.CYLN(R5),RTYCYL ;STORE RETRY CYLINDER
7365  034364  016537  000010  001634        MOV     P.BALO(R5),RTYBA ;STORE RETRY BUS ADDRESS
7366  034372  016537  000012  001636        MOV     P.WC(R5),RTYWC  ;STORE RETRY WORD COUNT
7367  034400  005437  001636                NEG     RTYWC           ;MAKE IT POSITIVE
```

```
7368  034404  022737  000400  001636          CMP     #256.,RTYWC     ;CHECK IF LESS THAN OR EQUAL TO A SECTOR
7369  034412  103043                          BHIS    15$             ;YES, CHECK IF ECC CORRECTION
7370  034414  016537  000026  001640          MOV     P.DTS(R5),RTYSEC ;STORE PRESENT TRACK AND SECTOR
7371  034422  016537  000030  001642          MOV     P.DCYL(R5),RTYCYL ;STORE PRESENT CYLINDER
7372  034430  105737  001640                  TSTB    RTYSEC          ;CHECK IF SECTOR = 0
7373  034434  001403                          BEQ     11$             ;YES, DECREMENT TRACK
7374  034436  105337  001640                  DECB    RTYSEC          ;DECREMENT SECTOR
7375  034442  000416                          BR      13$             ;GO CALCULATE WORD COUNT
7376
7377  034444  112737  000025  001640  11$:    MOVB    #21.,RTYSEC     ;SET SECTOR = 21
7378  034452  105737  001641                  TSTB    RTYTRK          ;CHECK IF TRACK = 0
7379  034456  001403                          BEQ     12$             ;YES, DECREMENT CYLINDER
7380  034460  105337  001641                  DECB    RTYTRK          ;DECREMENT TRACK
7381  034464  000405                          BR      13$             ;GO CALCULATE WORD COUNT
7382
7383  034466  112737  000002  001641  12$:    MOVB    #2,RTYTRK       ;SET TRACK = 2
7384  034474  005337  001642                  DEC     RTYCYL          ;DECREMENT CYLINDER
7385  034500  004737  037210          13$:    JSR     PC,CALWC        ;CALCULATE WORD COUNT
7386  034504  022737  000400  001636          CMP     #256.,RTYWC     ;CHECK IF LESS THAN OR EQUAL TO ONE SECTOR
7387  034512  103003                          BHIS    15$             ;YES, PERFORM ECC CORRECTION
7388  034514  012737  000400  001636          MOV     #256.,RTYWC     ;MAKE IT ONE SECTOR
7389  034522  004037  030654          15$:    JSR     R0,PRI200       ;PRINT HEADER
7390  034526  062364                          ERR048                  ;PRINT DATA CHECK
7391  034530  105765  000120                  TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERRROR RECOVERY
7392  034534  001123                          BNE     40$             ;YES, INDICATE UNSUCESSFUL RECOVERY
7393  034536  122765  000121  000123          CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
7394  034544  001101                          BNE     30$             ;NO, DO NOT DO ECC CORRECTION
7395  034546  032765  000100  000034          BIT     #ECH,P.ER(R5)   ;CHECK IF CORRECTABLE
7396  034554  001075                          BNE     30$             ;NO, DO NOT DO ECC CORRECTION
7397  034556  022765  010040  000060          CMP     #10040,P.EPOS(R5) ;CHECK IF ECC POSITION TOO LARGE
7398  034564  101406                          BLOS    16$             ;YES, REPORT ERROR
7399  034566  005765  000060                  TST     P.EPOS(R5)      ;CHECK IF ECC POSITION = 0
7400  034572  001403                          BEQ     16$             ;YES, REPORT ERROR
7401  034574  005765  000062                  TST     P.EPAT(R5)      ;CHECK IF ECC PATTERN = 0
7402  034600  001004                          BNE     17$             ;NO, DO ECC CORRECTION
7403  034602  004037  037322          16$:    JSR     R0,CONERR       ;TYPE "ECC LOGIC ERROR"
7404  034606  063130                          ERR066
7405  034610  000207                          RTS     PC              ;RETURN
7406
7407  034612  004737  026160          17$:    JSR     PC,ECCCOR       ;DO ECC CORRECTION
7408  034616  105765  000210                  TSTB    P.ASSN(R5)      ;CHECK IF IN ASSIGNMENT SEQUENCE
7409  034622  001017                          BNE     18$             ;YES, DO NOT CHECK ECC CORRECTION
7410  034624  013765  001634  000214          MOV     RTYBA,P.ERHD(R5) ;LOAD START OF SECTOR FOR
7411                                                                  ;   DATA COMPARISON
7412  034632  013765  001636  000222          MOV     RTYWC,P.BFCP(R5) ;LOAD NUMBER OF WORDS OF
7413  034640  013765  001636  000220          MOV     RTYWC,P.CMLG(R5) ;   SECTOR PRESENT
7414  034646  112765  177777  000117          MOVB    #-1,P.ECMP(R5)  ;SET ECC COMPARE FLAG
7415  034654  004737  023362                  JSR     PC,CMPBUF       ;CHECK ECC CORRECTION
7416  034660  034670                          20$                     ;INDICATE ECC MISCORRECTION
7417  034662  004737  036276          18$:    JSR     PC,SURECY       ;INDICATE SUCCESSFUL RECOVERY
7418  034666  000207                          RTS     PC              ;RETURN
7419
7420  034670  032777  020000  144242  20$:    BIT     #SW13,@SWR      ;CHECK IF INHIBIT ERROR PRINT OUT
7421  034676  001024                          BNE     30$             ;YES, GO TO RECOVERY SEQUENCE
7422  034700  104401  063024                  TYPE    ,ERR060         ;TYPE "ECC MISCORRECTION"
7423  034704  104401  064303                  TYPE    ,ERR211         ;TYPE "RKPOS"
```

```
7424  034710  016546  000060                  MOV     P.EPOS(R5),-(SP) ;STORE RKPOS FOR PRINT OUT
7425  034714  004737  052774                  JSR     PC,BINOCT       ;CONVERT TO OCTAL
7426  034720  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
7427  034724  104401  064312                  TYPE    ,ERR212         ;TYPE "RKPAT"
7428  034730  016546  000062                  MOV     P.EPAT(R5),-(SP) ;STORE RKPAT FOR PRINT OUT
7429  034734  004737  052774                  JSR     PC,BINOCT       ;COVERT TO OCTAL
7430  034740  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
7431  034744  004737  036024                  JSR     PC,PRERWD       ;PRINT WORD IN ERROR
7432  034750  105265  000147          30$:    INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7433  034754  122765  000043  000147          CMPB    #43,P.RERD(R5)  ;CHECK IF UNSUCCESSFUL RETRY
7434  034762  101410                          BLOS    40$             ;YES, INDICATE UNSUCESSFUL RETRY
7435  034764  122765  000121  000123          CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA
7436  034772  001407                          BEQ     45$             ;YES, CHECK FOR OFFSET
7437  034774  122765  000021  000147          CMPB    #21,P.RERD(R5)  ;CHECK IF ABOUT TO ENTER OFFSET OPERATATIONS
7438  035002  101021                          BHI     50$             ;NO, DO REREAD
7439  035004  004737  031502          40$:    JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
7440  035010  000207                          RTS     PC              ;RETURN
7441
7442  035012  122765  000021  000147  45$:    CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET +400 MICRO-INCHES
7443  035020  001007                          BNE     48$             ;NO, GO DETERMINE IF OFFSET IS TO
7444                                                                  ;  BE ISSUED
7445  035022  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
7446  035024  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER AS INDEX
7447  035030  146437  001514  001513          BICB    INTMSK(R4),O.OVER ;RESET ISSUE IMPLIED SEEK
7448  035036  012604                          MOV     (SP)+,R4        ;RESTORE R4
7449  035040  004037  037026          48$:    JSR     R0,DETOFF       ;DETERMINE IF OFFSET IS TO BE ISSUED
7450  035044  035134                          60$                     ;OFFSET ISSUED RETURN
7451  035046  116565  000123  000001  50$:    MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST COMMAND
7452  035054  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK
7453  035062  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER
7454  035070  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
7455  035076  013765  001636  000012          MOV     RTYWC,P.WC(R5)  ;LOAD WORD COUNT
7456  035104  005465  000012                  NEG     P.WC(R5)        ;MAKE IT NEGATIVE
7457  035110  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
7458  035116  001003                          BNE     55$             ;NO, GO ISSUE COMMAND
7459  035120  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
7460  035126  004037  051406          55$:    JSR     R0,Q.PUSH       ;PUT PARAMETER BLOCK ON COMMAND
7461  035132  001324                          CINITQ                  ;COMMAND INITIATION QUEUE
7462  035134  000207                  60$:    RTS     PC              ;RETURN
7463
7464  035136  032765  040000  000020  E5$:    BIT     #WCE,P.CS2(R5)  ;CHECK IF WRITE CHECK ERROR
7465  035144  001002                          BNE     1$              ;YES, PROCESS WRITE CHECK ERROR
7466  035146  000137  035310                  JMP     E10$            ;NO, CONTINUE
7467
7468  035152  004037  030654          1$:     JSR     R0,PRI200       ;PRINT WRITE CHECK ERROR
7469  035156  062376                          ERR050
7470  035160  105765  000120                  TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR CORRECTION AND RETRY
7471  035164  001006                          BNE     5$              ;YES, INITIATE NEXT COMMAND
7472  035166  105265  000147                  INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7473  035172  122765  000021  000147          CMPB    #21,P.RERD(R5)  ;CHECK IF 16 REREADS OCCURRED
7474  035200  101003                          BHI     10$             ;NO, START RECOVERY SEQUENCE
7475  035202  004737  031502          5$:     JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
7476  035206  000207                          RTS     PC              ;RETURN
7477
7478  035210  016537  000024  001634  10$:    MOV     P.BAR(R5),RTYBA ;GET ADDRESS OF WORD IN ERROR
7479  035216  162737  000002  001634          SUB     #2,RTYBA        ;ADJUST ADDRESS
```

```
7480  035224  004737  035612                  JSR     PC,CALSEC       ;CALCULATE SECTOR, TRACK AND CYLINDER
7481  035230  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;GET PREVIOUS COMMAND
7482  035236  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;GET SECTOR AND TRACK
7483  035244  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;GET CYLINDER
7484  035252  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;GET BUS ADDRESS
7485  035260  013765  001636  000012          MOV     RTYWC,P.WC(R5)  ;GET WORD COUNT
7486  035266  005465  000012                  NEG     P.WC(R5)        ;MAKE IT NEGATIVE
7487  035272  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;SET ISSUE WRITE BEFORE WRITE CHECK
7488  035300  004037  051406                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER IN
7489  035304  001324                          CINITQ                  ;  COMMAND INITIATION QUEUE
7490  035306  000207                          RTS     PC              ;RETURN
7491
7492
7493  035310  032765  000004  000212  E10$:   BIT     #BIT2,P.ERR(R5) ;CHECK IF DATA COMPARE ERROR
7494  035316  001001                          BNE     2$              ;YES, CONTINUE PROCESSING ERROR
7495  035320  000000                          HALT                    ;  ### UNKNOWN ENTRY
7496  035322  005046                  2$:     CLR     -(SP)           ;MAKE ROOM ON STACK
7497  035324  116516  000217                  MOVB    P.ERAD(R5),(SP) ;GET WORD IN ERROR
7498  035330  006316                          ASL     (SP)            ;MULTIPLY BY 2
7499  035332  066516  000214                  ADD     P.ERHD(R5),(SP) ;CALCULATE ADDRESS OF MISCOMPARED WORD
7500  035336  012637  001634                  MOV     (SP)+,RTYBA     ;LOAD ADDRESS OF BAD WORD
7501  035342  004737  035612                  JSR     PC,CALSEC       ;CALCULATE CYLINDER, TRACK, AND SECTOR
7502  035346  032765  000002  000150          BIT     #BIT1,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR HAS ALREADY OCCURRED
7503  035354  001037                          BNE     10$             ;YES, GO TO RETRY SEQUENCE
7504  035356  052765  000002  000150          BIS     #BIT1,P.DSTT(R5) ;SET DATA TYPE ERROR OCCURRED
7505  035364  100003                          BPL     3$              ;CHECK IF FIRST ERROR
7506  035366  052765  000001  000150          BIS     #BIT0,P.DSTT(R5) ;SET INITIAL DATA ERROR
7507  035374  004737  030000          3$:     JSR     PC,PRI000       ;PRINT HEADER
7508  035400  032777  020000  143532          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
7509  035406  001013                          BNE     5$              ;YES, CHECK IF HALT ON ERROR
7510  035410  104401  063050                  TYPE    ,ERR062         ;TYPE DATA COMPARE ERROR
7511  035414  005765  000150                  TST     P.DSTT(R5)      ;CHECK IF FIRST ERROR
7512  035420  100402                          BMI     4$              ;YES, GO PRINT ERROR WORD
7513  035422  104401  063721                  TYPE    ,ERR202         ;TYPE ERROR DURING ERROR RECOVERY
7514  035426  004737  036024          4$:     JSR     PC,PRERWD       ;PRINT WORD IN ERROR
7515  035432  004737  030244                  JSR     PC,PRI002       ;PRINT CURRENTLY SUPPLIED COMMAND
7516  035436  042765  100200  000150  5$:     BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
7517  035444  004737  027442                  JSR     PC,PRIBUF       ;GO PRINT BUFFER
7518  035450  004737  031250                  JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
7519  035454  105765  000120          10$:    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR CORRECTION AND RECOVERY
7520  035460  001006                          BNE     15$             ;YES, INDICATE UNSUCCESSFUL RECOVERY
7521  035462  105265  000147                  INCB    P.RERD(R5)      ;INCREMENT RETRY COUNT
7522  035466  122765  000043  000147          CMPB    #43,P.RERD(R5)  ;CHECK IF ERROR RECOVERY UNSUCCESSFUL
7523  035474  101003                          BHI     20$             ;NO, GO TO RECOVERY SEQUENCE
7524  035476  004737  031502          15$:    JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
7525  035502  000207                          RTS     PC              ;RETURN
7526
7527  035504  122765  000021  000147  20$:    CMPB    #21,P.RERD(R5)  ;CHECK IF ENTERING OFFSET
7528  035512  001007                          BNE     25$             ;NO, DETERMINE IF TIME TO CHANGE OFFSET
7529  035514  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
7530  035516  116504  000000                  MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
7531  035522  146437  001514  001513          BICB    INTMSK(R4),O.OVER ;RESET SEEK BEFORE DATA TRANSFER
7532  035530  012604                          MOV     (SP)+,R4        ;RESTORE R4
7533  035532  004037  037026          25$:    JSR     R0,DETOFF       ;DETERMINE IF OFFSET TO BE ISSUED
7534  035536  035610                          50$                     ;OFFSET HAS BEEN ISSUED
7535  035540  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;GET LAST ISSUED COMMAND
```

```
 7536  035546  013765  001640  000004          MOV     RTYSEC,P.SECT(R5) ;LOAD SECTOR AND TRACK ADDRESS
 7537  035554  013765  001642  000002          MOV     RTYCYL,P.CYLN(R5) ;LOAD CYLINDER ADDRESS
 7538  035562  013765  001634  000010          MOV     RTYBA,P.BALO(R5) ;LOAD BUS ADDRESS
 7539  035570  013765  001636  000012          MOV     RTYWC,P.WC(R5)   ;GET WORD COUNT
 7540  035576  005465  000012                  NEG     P.WC(R5)         ;MAKE IT NEGATIVE
 7541  035602  004037  051406                  JSR     R0,Q.PUSH        ;ENQUEUE PARAMETER BLOCK IN COMMAND
 7542  035606  001324                          CINITQ                   ;  INITIATION QUEUE
 7543  035610  000207                  50$:    RTS     PC               ;RETURN
```

```
                                        .SBTTL  CALCULATE CYLINDER, TRACK, AND SECTOR FOR RETRY
                                
035612  010146                  CALSEC: MOV     R1,-(SP)        ;STORE R1 ON STACK
035614  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
035616  013703  001634                  MOV     RTYBA,R3        ;GET ADDRESS OF ERROR WORD
035622  166503  000010                  SUB     P.BALO(R5),R3   ;CALCULATE BYTES CORRECTLY TRANSFERRED
035626  042703  000777                  BIC     #777,R3         ;MAKE IT ON SECTOR BOUNDARY
035632  010337  001634                  MOV     R3,RTYBA        ;STORE CALCULATED BUS ADDRESS
035636  066537  000010  001634          ADD     P.BALO(R5),RTYBA ;CALCULATE START OF SECTOR WITH ERROR
035644  016501  000012                  MOV     P.WC(R5),R1     ;GET INITIAL WORD COUNT
035650  005401                          NEG     R1              ;MAKE IT POSITIVE
035652  006203                          ASR     R3              ;DETERMINE WORDS BEFORE ERROR
035654  160301                          SUB     R3,R1           ;CALCULATE NUMBER OF WORDS LEFT
035656  022701  000400                  CMP     #256.,R1        ;CHECK IF LESS THAN OR EQUAL TO ONE SECTOR
035662  103002                          BHIS    5$              ;LOAD RETRY WORD COUNT
035664  012701  000400                  MOV     #256.,R1        ;NO, MAKE IT ONE SECTOR
035670  010137  001636          5$:     MOV     R1,RTYWC        ;LOAD RETY WORD COUNT
035674  116537  000005  001641          MOVB    P.TRCK(R5),RTYTRK ;STORE RETRY TRACK
035702  016537  000002  001642          MOV     P.CYLN(R5),RTYCYL ;STORE RETRY CYLINDER
035710  010346                          MOV     R3,-(SP)        ;STORE WORDS TRANSFERRED
035712  116616  000001                  MOVB    1(SP),(SP)      ;STORE NUMBER OF SECTORS
035716  105066  000001                  CLRB    1(SP)           ;CLEAR HIGH ORDER BYTE
035722  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
035724  116516  000004                  MOVB    P.SECT(R5),(SP) ;GET INITIAL SECTOR
035730  062616                          ADD     (SP)+,(SP)      ;ADD SECTORS TRANSFERRED
035732  022716  000025          10$:    CMP     #21.,(SP)       ;CHECK IF SECTOR COUNT GREATER THAN 21
035736  103005                          BHIS    15$             ;NO, CHECK TRACK
035740  162716  000026                  SUB     #22.,(SP)       ;SUBTRACT 22 SECTORS/TRACK
035744  105237  001641                  INCB    RTYTRK          ;INCREMENT TRACK
035750  000770                          BR      10$             ;CHECK SECTOR COUNT

035752  112637  001640          15$:    MOVB    (SP)+,RTYSEC    ;STORE RETRY SECTOR
035756  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
035760  113716  001641                  MOVB    RTYTRK,(SP)     ;STORE RETRY TRACK
035764  122716  000002          17$:    CMPB    #2,(SP)         ;CHECK TRACK GREATER THAN 2
035770  103005                          BHIS    20$             ;NO, RETURN
035772  162716  000003                  SUB     #3,(SP)         ;SUBTRACT 3 TRACKS/CYLINDER
035776  005237  001642                  INC     RTYCYL          ;INCREMENT CYLINDER
036002  000770                          BR      17$             ;CHECK TRACK COUNT

036004  112637  001641          20$:    MOVB    (SP)+,RTYTRK    ;LOAD RETRY COUNT
036010  112737  000001  001644          MOVB    #1,RTYSCN       ;INDICATE ONE SECTOR IN RETRY COUNT
036016  012603                          MOV     (SP)+,R3        ;RESTORE R3
036020  012601                          MOV     (SP)+,R1        ;RESTORE R1
036022  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL  PRINT FIRST ERROR WORD IN BUFFER
                                
036024  010046          PRERWD: MOV     R0,-(SP)        ;STORE R0 ON STACK
036026  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
036030  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
036032  010446                  MOV     R4,-(SP)        ;STORE R4 ON STACK
036034  104401  061173          TYPE    ,ERR012         ;TYPE ERROR HEADER
036040  016503  000214          MOV     P.ERHD(R5),R3   ;GET ADDRESS OF START OF SECTOR
036044  005046                  CLR     -(SP)           ;MAKE ROOM ON STACK
036046  116516  000217          MOVB    P.ERAD(R5),(SP) ;GET WORD IN ERROR OF SECTOR
036052  006316                  ASL     (SP)            ;MULTIPLY IT BY 2
036054  011601                  MOV     (SP),R1         ;STORE IT IN R1
036056  062603                  ADD     (SP),R3         ;GET ADDRESS OF BAD WORD
036060  010346                  MOV     R3,-(SP)        ;STORE ADDRESS FOR PRINT OUT
036062  004737  052774          JSR     PC,BINOCT       ;CONVERT TO OCTAL
036066  122765  000001  000217  CMPB    #1,P.ERAD(R5)   ;CHECK IF IN DESCRIPTOR AREA
036074  103403                  BLO     5$              ;NO, CALCULATE WORD
036076  104401  060515          TYPE    ,QUESMK         ;TYPE QUESTION MARKS
036102  000436                  BR      10$             ;GO TYPE WORD READ
                                
036104  016504  000214  5$:     MOV     P.ERHD(R5),R4   ;GET HEADER FOR PATTERN INDEX
036110  016400  000002          MOV     2(R4),R0        ;GET NUMBER OF WORDS WRITTEN IN SECTOR
036114  042700  177400          BIC     #177400,R0      ;KEEP LOW BYTE
036120  005200                  INC     R0              ;INCREMENT R0 TO MAKE OFFSET FROM START
                                                        ;  OF SECTOR
036122  126500  000217          CMPB    P.ERAD(R5),R0   ;CHECK IF IN ZERO FILL AREA
036126  101402                  BLOS    7$              ;NO, CALCULATF ADDRESS OF
                                                        ;  WORD IN DATA PATTERN
036130  005046                  CLR     -(SP)           ;LOAD ZERO FOR PRINT OUT
036132  000414                  BR      9$              ;GO PRINT EXPECTED CONTENTS
                                
036134  011404          7$:     MOV     (R4),R4         ;GET DATA PATTERN NUMBER FOR INDEX
036136  042704  177760          BIC     #177760,R4      ;MASK OFF INSIGNIFICANT BITS
036142  006304                  ASL     R4              ;MULTIPLY BY 2
036144  016404  004602          MOV     PATTAB(R4),R4   ;GET PATTERN BASE
036150  162701  000004          SUB     #4,R1           ;SUBTRACT 4 FROM SECTOR OFFSET
036154  042701  177700          BIC     #177700,R1      ;MAKE IT MOD 32 WORDS
036160  060104                  ADD     R1,R4           ;CALCULATE PATTERN ADDRESS
036162  011446                  MOV     (R4),-(SP)      ;STORE EXPECTED CONTENTS FOR PRINT OUT
036164  104401  060530  9$:     TYPE    ,BLANKS         ;TYPE BLANKS
036170  004737  052774          JSR     PC,BINOCT       ;CONVERT TO OCTAL
036174  104401  060530          TYPE    ,BLANKS         ;TYPE BLANKS
036200  011346          10$:    MOV     (R3),-(SP)      ;STORE WORD READ
036202  004737  052774          JSR     PC,BINOCT       ;CONVERT TO OCTAL
036206  104401  060530          TYPE    ,BLANKS         ;TYPE BLANKS
036212  005046                  CLR     -(SP)           ;MAKE ROOM ON STACK
036214  116516  000217          MOVB    P.ERAD(R5),(SP) ;STORE SECTOR WORD NUMBER FOR PRINT OUT
036220  004737  052774          JSR     PC,BINOCT       ;CONVERT TO OCTAL
036224  105765  000217          TSTB    P.ERAD(R5)      ;CHECK IF ERROR IS IN PATTERN WORD
036230  001003                  BNE     15$             ;NO, PRINT PATTERN NUMBER
036232  104401  060515          TYPE    ,QUESMK         ;TYPE QUESTION MARKS
036236  000410                  BR      20$             ;GO RESTORE REGISTERS
                                
036240  104401  060530  15$:    TYPE    ,BLANKS         ;TYPE BLANKS
036244  017546  000214          MOV     @P.ERHD(R5),-(SP) ;GET FIRST WORD OF SECTOR READ
036250  042716  177760          BIC     #177760,(SP)    ;CLEAR INSIGNIFICANT BITS
```

```
 7645  036254  004737  052774          JSR     PC,BINOCT       ;CONVERT TO OCTAL
 7646  036260  104401  001165  20$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
 7647  036264  012604                  MOV     (SP)+,R4        ;RESTORE R4
 7648  036266  012603                  MOV     (SP)+,R3        ;RESTORE R3
 7649  036270  012601                  MOV     (SP)+,R1        ;RESTORE R1
 7650  036272  012600                  MOV     (SP)+,R0        ;RESTORE R0
 7651  036274  000207                  RTS     PC              ;RETURN
```

```
                                        .SBTTL  SUCCESSFUL RECOVERY CLEAN UP
                                
036276  032765  000001  000150  SURECY: BIT     #BIT0,P.DSTT(R5) ;CHECK IF DATA TYPE ERROR
036304  001466                          BEQ     10$              ;NO, CLEAN UP FOR NEXT COMMAND
036306  105765  000147                  TSTB    P.RERD(R5)       ;CHECK IF FIRST ERROR
036312  001003                          BNE     5$               ;NO, CHECK IF OFFSET
036314  005265  000176                  INC     P.SECC(R5)       ;INCREMENT NO. OF ECC RECOVERABLE ERRORS
036320  000426                          BR      9$               ;GO CLEAN UP FOR NEXT COMMAND
                                
036322  122765  000020  000147  5$:     CMPB    #20,P.RERD(R5) ;CHECK IF OFFSET
036330  103403                          BLO     8$             ;YES, INCREMENT OFFSET RECOVERABLE
036332  005265  000172                  INC     P.SRRD(R5)     ;INCREMENT REREAD RECOVERABLE
036336  000417                          BR      9$             ;GO CLEAN UP FOR NEXT COMMAND
                                
036340  005265  000174          8$:     INC     P.SOFF(R5)     ;INCREMENT OFFSET RECOVERABLE
036344  032777  020000  142566          BIT     #SW13,@SWR     ;CHECK IF INHIBIT PRINT OUT
036352  001011                          BNE     9$             ;YES, CLEAN UP FOR NEXT COMMAND
036354  104401  063552                  TYPE    ,ERR108        ;TYPE "OFFSET"
036360  005046                          CLR     -(SP)          ;MAKE ROOM ON STACK
036362  116516  000006                  MOVB    P.OFST(R5),(SP) ;STORE OFFSET FOR PRINT OUT
036366  004737  052774                  JSR     PC,BINOCT      ;CONVERT TO OCTAL
036372  104401  001165                  TYPE    ,$CRLF         ;TYPE <CR><LF>
036376  105037  001644          9$:     CLRB    RTYSCN         ;CLEAR SECTOR RETRY FLAG
036402  016546  000176                  MOV     P.SECC(R5),-(SP) ;STORE NUMBER OF ECC RECOVERABLE ERRORS
036406  066516  000172                  ADD     P.SRRD(R5),(SP) ;ADD NUMBER OF REREAD RECOVERABLE ERRORS
036412  066516  000174                  ADD     P.SOFF(R5),(SP) ;ADD NUMBER OF OFFSET RECOVERABLE ERRORS
036416  022665  000100                  CMP     (SP)+,P.CERT(R5) ;CHECK IF SOFT ERROR RATE EXCEEDED
036422  101417                          BLOS    10$             ;NO, GO ISSUE NEXT COMMAND
036424  032777  040020  142506          BIT     #SW4!SW14,@SWR  ;CHECK IF INHIBIT AUTOMATIC DRIVE DROPPING
036432  001013                          BNE     10$             ;YES, GO ISSUE NEXT COMMAND
036434  004737  022642                  JSR     PC,BUFREL       ;GO RELEASE BUFFER
036440  104401  064047                  TYPE    ,ERR205         ;TYPE "ERROR THRESHOLD EXCEEDED"
036444  004737  013004                  JSR     PC,DROP         ;GO DROP DRIVE
036450  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENTLY BEING PROCESSED
036454  004737  022342                  JSR     PC,GETBUF       ;GET BUFFER FOR NEXT COMMAND
036460  000207                          RTS     PC              ;RETURN
                                
036462  032777  020000  142450  10$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
036470  001002                          BNE     15$             ;YES, CHECK IF READ PACK SERIAL NUMBER
036472  104401  063757                  TYPE    ,ERR203         ;TYPE RECOVERABLE ERROR
036476  105037  001644          15$:    CLRB    RTYSCN          ;CLEAR SECTOR RETRY COUNT
036502  005065  000146                  CLR     P.RECT(R5)      ;CLEAR ERROR FLAGS AND RETRY COUNTS
036506  005065  000150                  CLR     P.DSTT(R5)
036512  005065  000212                  CLR     P.ERR(R5)
036516  005037  001566                  CLR     ERRPRO          ;CLEAR ERROR CURRENTLY BEING PROCESSED
036522  010446                          MOV     R4,-(SP)        ;STORE R4 ON STACK
036524  116504  000000                  MOVB    P.DRVN(R5),R4   ;STORE DRIVE NUMBER FOR INDEX
036530  156437  001514  001513          BISB    INTMSK(R4),O.OVER ;ALLOW OVERLAPPED OPERATIONS ON THIS DRIVE
036536  012604                          MOV     (SP)+,R4        ;RESTORE R4
036540  105765  000210                  TSTB    P.ASSN(R5)      ;CHECK IF IN ASSIGNMENT SEQUENCE
036544  001025                          BNE     20$             ;YES, TERMINATE SEQUENCE
036546  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS CYLINDER
036554  001410                          BEQ     16$             ;YES, GO ISSUE RELEASE
036556  122765  000140  000001          CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE
036564  001013                          BNE     17$             ;NO, CARRY OUT NORMAL RECOVERY
036566  004037  051406                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN DRIVE
```

```
7708  036572  001314                          AVAILQ                          ;  AVAILIABLE QUEUE
7709  036574  000207                          RTS     PC                      ;RETURN
7710
7711  036576  112765  000140  000001  16$:    MOVB    #RELEAS,P.CMND(R5) ;ISSUE RELEASE COMMAND
7712  036604  004037  051406                  JSR     R0,Q.PUSH               ;ENQUEUE PARAMETER BLOCK IN COMMAND
7713  036610  001324                          CINITQ                          ;  INITIATION QUEUE
7714  036612  000207                          RTS     PC                      ;RETURN
7715
7716  036614  000137  024616          17$:    JMP     NORM1                   ;CARRY OUT NORMAL DATA TRANSFER TERMINATION
7717
7718                                  ;       CHECK IF ANY OF THE FOLLOWING ASSIGNMENT STAGES
7719                                  ;               READ DRIVE STATUS
7720                                  ;               START SPINDLE
7721                                  ;               SET VOLUME VALID
7722                                  ;               RECAL
7723                                  ;               READ PACK SERIAL NUMBER
7724  036620  132765  000117  000210  20$:    BITB    #BIT0!BIT1!BIT2!BIT3!BIT6,P.ASSN(R5)
7725  036626  001402                          BEQ     25$                     ;NO, PROCESS WRITE PACK
7726  036630  000137  013504                  JMP     ASNORM                  ;PROCESS NORMAL TERMINATION
7727
7728  036634  032765  002000  000014  25$:    BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
7729  036642  001030                          BNE     30$                     ;YES, GO DROP DRIVE
7730  036644  105065  000211                  CLRB    P.SEEK(R5)              ;CLEAR SEEK TO PREVIOUS LOCATION
7731  036650  116565  000123  000001          MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE LAST WRITE/WRITE CHECK COMMAND
7732  036656  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;SET ISSUE WRITE BEFORE WRITE CHECK
7733  036664  016565  000126  000004          MOV     P.RSEC(R5),P.SECT(R5) ;LOAD PREVIOUS TRACK AND SECTOR
7734  036672  016565  000124  000002          MOV     P.RCYL(R5),P.CYLN(R5) ;LOAD PREVIOUS CYLINDER
7735  036700  016565  000132  000010          MOV     P.RBAL(R5),P.BALQ(R5) ;LOAD PREVIOUS BUS ADDRESS
7736  036706  016565  000130  000012          MOV     P.RWC(R5),P.WC(R5) ;LOAD PREVIOUS WORD COUNT
7737  036714  004037  051406                  JSR     R0,Q.PUSH               ;ENQUEUE PARAMETER BLOCK IN COMMAND
7738  036720  001324                          CINITQ                          ;  INITIATION QUEUE
7739  036722  000207                          RTS     PC                      ;RETURN
7740
7741  036724  004737  022642          30$:    JSR     PC,BUFREL               ;GO RELEASE BUFFER
7742  036730  104401  056260                  TYPE    ,OPR008                 ;TYPE "OPERATOR INITIATED DROP DRIVE
7743                                                                          ;  DURING PACK WRITING"
7744  036734  004737  013004                  JSR     PC,DROP                 ;GO DROP DRIVE
7745  036740  004737  022342                  JSR     PC,GETBUF               ;GET BUFFER FOR NEXT COMMAND
7746  036744  000207                          RTS     PC                      ;RETURN
```

```
                                        .SBTTL  GET CURRENT STATUS INFORMATION

036746  010446                  GETCUR: MOV     R4,-(SP)        ;STORE R4 ON STACK
036750  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
036752  010504                          MOV     R5,R4           ;LOAD START OF CURRENT PARRAMETER BLOCK
036754  012703  004516                  MOV     #PARM10,R3      ;LOAD START OF RETRY PARAMETER BLOCK

036760  012423                  5$:     MOV     (R4)+,(R3)+     ;COPY PARAMTER BLOCK
036762  022703  004602                  CMP     #PARM10+P.EPAT+2,R3 ;CHECK IF FINISHED
036766  001374                          BNE     5$              ;NO, GET NEXT WORD
036770  012603                          MOV     (SP)+,R3        ;RESTORE R3
036772  012604                          MOV     (SP)+,R4        ;RESTORE R4
036774  000207                          RTS     PC              ;RETURN

                                        .SBTTL  RESTORE STATUS

036776  010446                  RSTAT:  MOV     R4,-(SP)        ;STORE R4 ON STACK
037000  010346                          MOV     R3,-(SP)        ;STORE R3 ON STACK
037002  010504                          MOV     R5,R4           ;LOAD START OF CURRENT PARRAMETER BLOCK
037004  012703  004516                  MOV     #PARM10,R3      ;LOAD START OF RETRY PARAMETER BLOCK

037010  012324                  5$:     MOV     (R3)+,(R4)+     ;COPY PARAMTER BLOCK
037012  022703  004602                  CMP     #PARM10+P.EPAT+2,R3 ;CHECK IF FINISHED
037016  001374                          BNE     5$              ;NO, GET NEXT WORD
037020  012603                          MOV     (SP)+,R3        ;RESTORE R3
037022  012604                          MOV     (SP)+,R4        ;RESTORE R4
037024  000207                          RTS     PC              ;RETURN
```

```
7774                                        .SBTTL  DETERMINE OFFSET FOR RETRY
7775
7776  037026  122765  000021  000147  DETOFF: CMPB    #21,P.RERD(R5)  ;CHECK IF OFFSET = +400 MICRO-INCHES
7777  037034  001004                          BNE     10$             ;NO, CONTINUE
7778  037036  112765  000020  000006          MOVB    #OFFP4,P.OFST(R5) ;LOAD OFFSET VALUE = +400 MICRO-INCHES
7779  037044  000447                          BR      15$             ;ISSUE OFFSET
7780
7781  037046  122765  000024  000147  10$:     CMPB    #24,P.RERD(R5)  ;CHECK IF OFFSET = -400 MICRO-INCHES
7782  037054  001004                          BNE     11$             ;NO, CONTINUE
7783  037056  112765  000220  000006          MOVB    #OFFM4,P.OFST(R5) ;LOAD OFFSET VALUE = -400 MICRO-INCHES
7784  037064  000437                          BR      15$             ;ISSUE OFFSET
7785
7786  037066  122765  000027  000147  11$:     CMPB    #27,P.RERD(R5)  ;CHECK IF OFFSET = +800 MICRO-INCHES
7787  037074  001004                          BNE     12$             ;NO, CONTINUE
7788  037076  112765  000040  000006          MOVB    #OFFP8,P.OFST(R5) ;LOAD OFFSET VALUE = +800 MICRO-INCHES
7789  037104  000427                          BR      15$             ;ISSUE OFFSET
7790
7791  037106  122765  000032  000147  12$:     CMPB    #32,P.RERD(R5)  ;CHECK IF OFFSET = -800 MICRO-INCHES
7792  037114  001004                          BNE     13$             ;NO, CONTINUE
7793  037116  112765  000240  000006          MOVB    #OFFM8,P.OFST(R5) ;LOAD OFFSET VALUE = -800 MICRO-INCHES
7794  037124  000417                          BR      15$             ;ISSUE OFFSET
7795
7796  037126  122765  000035  000147  13$:     CMPB    #35,P.RERD(R5)  ;CHECK IF OFFSET = +1200 MICRO-INCHES
7797  037134  001004                          BNE     14$             ;NO, CONTINUE
7798  037136  112765  000060  000006          MOVB    #OFFP12,P.OFST(R5) ;LOAD OFFSET VALUE = +1200 MICRO-INCHES
7799  037144  000407                          BR      15$             ;ISSUE OFFSET
7800
7801  037146  122765  000040  000147  14$:     CMPB    #40,P.RERD(R5)  ;CHECK IF OFFSET = -1200 MICRO-INCHES
7802  037154  001013                          BNE     30$             ;NO, DO REREAD
7803  037156  112765  000260  000006          MOVB    #OFFM12,P.OFST(R5) ;LOAD OFFSET VALUE = -1200 MICRO-INCHES
7804  037164  112765  000115  000001  15$:     MOVB    #OFFSET,P.CMND(R5) ;LOAD OFFSET COMMAND
7805  037172  004037  051406                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
7806  037176  001324                          CINITQ                  ;  INITIATION QUEUE
7807  037200  011000                          MOV     (R0),R0         ;LOAD OFFSET ISSUED RETURN
7808  037202  000200                          RTS     R0              ;RETURN
7809
7810  037204  005720                  30$:     TST     (R0)+           ;ADJUST RETURN FOR NO OFFSET
7811  037206  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL  CALCULATE RETRY WORD COUNT

7812
7813
7814  037210  013746  001642    CALWC:  MOV     RTYCYL,-(SP)        ;STORE RETRY CYLINDER
7815  037214  166516  000002            SUB     P.CYLN(R5),(SP)     ;SUBTRACT INITIAL CYLINDER
7816  037220  011646                    MOV     (SP),-(SP)          ;STORE CYLINDER DIFFERENCE
7817  037222  006316                    ASL     (SP)                ;MULTIPLY DIFFERENCE BY 2
7818  037224  062616                    ADD     (SP)+,(SP)          ;MULTIPLY DIFFERENCE BY 3
7819  037226  005046                    CLR     -(SP)               ;MAKE ROOM ON STACK
7820  037230  113716  001641            MOVB    RTYTRK,(SP)         ;GET RETRY TRACK
7821  037234  062616                    ADD     (SP)+,(SP)          ;ADD RETRY TRACK
7822  037236  005046                    CLR     -(SP)               ;MAKE ROOM ON THE STACK
7823  037240  116516  000005            MOVB    P.TRCK(R5),(SP)     ;GET INITIAL TRACK
7824  037244  162616                    SUB     (SP)+,(SP)          ;SUBTRACT INITIAL TRACK
7825  037246  012746  000026            MOV     #22.,-(SP)          ;MULTIPLY TRACK BY 22
7826  037252  004737  054714            JSR     PC,$MULT
7827  037256  012616                    MOV     (SP)+,(SP)          ;THROW AWAY MOST SIGNIFICANT BITS
7828  037260  005046                    CLR     -(SP)               ;MAKE ROOM ON THE STACK
7829  037262  113716  001640            MOVB    RTYSEC,(SP)         ;GET RETRY SECTOR
7830  037266  062616                    ADD     (SP)+,(SP)          ;ADD PRESENT SECTOR
7831  037270  005046                    CLR     -(SP)               ;MAKE ROOM ON STACK
7832  037272  116516  000004            MOVB    P.SECT(R5),(SP)     ;GET INITIAL SECTOR
7833  037276  162616                    SUB     (SP)+,(SP)          ;CALCULATE NUMBER OF SECTORS UNTIL ERROR
7834  037300  111666  000001            MOVB    (SP),1(SP)          ;MULTIPLY BY 256 WORDS/SECTOR
7835  037304  105016                    CLRB    (SP)
7836  037306  161637  001636            SUB     (SP),RTYWC          ;SUBTRACT FROM INITIAL WORD COUNT
7837  037312  006316                    ASL     (SP)                ;DETERMINE NUMBER OF BYTES
7838  037314  062637  001634            ADD     (SP)+,RTYBA         ;CALCULATE STARTING BUFFER ADDRESS
7839  037320  000207                    RTS     PC                  ;RETURN
```

```
                                        .SBTTL  RECOVERABLE CONTROLLER ERROR TERMINATION
                                
037322  012037  037344              CONERR: MOV     (R0)+,2$            ;STORE ERROR MESSAGE ADDRESS
037326  004737  030000                      JSR     PC,PRI000           ;PRINTER HEADER
037332  032777  020000  141600              BIT     #SW13,@SWR          ;CHECK IF INHIBIT PRINT OUT
037340  001032                              BNE     10$                 ;YES, CHECK IF HALT ON ERROR
037342  104401                              TYPE                        ;TYPE ERROR MESSAGE
037344  000000                      2$:     .WORD   0
037346  104401  001165                      TYPE    ,$CRLF              ;TYPE <CR><LF>
037352  004737  026074                      JSR     PC,PRISTT           ;PRINT CONTROLLER REGISTERS
037356  004737  030244                      JSR     PC,PRI002           ;PRINT CURRENT EXECUTED COMMAND
037362  022737  063130  037344              CMP     #ERR066,2$          ;CHECK IF ECC LOGIC ERROR
037370  001016                              BNE     10$                 ;NO, CHECK IF HALT ON ERROR
037372  104401  063150                      TYPE    ,ERR067             ;TYPE "RKECPT RKECPS"
037376  016546  000062                      MOV     P.EPAT(R5),-(SP)    ;STORE ECC PATTERN FOR PRINT OUT
037402  004737  052774                      JSR     PC,BINOCT           ;CONVERT TO OCTAL
037406  104401  060530                      TYPE    ,BLANKS             ;TYPE 2 SPACES
037412  016546  000060                      MOV     P.EPOS(R5),-(SP)    ;STORE ECC POSITION FOR PRINT OUT
037416  004737  052774                      JSR     PC,BINOCT           ;CONVERT TO OCTAL
037422  104401  001165                      TYPE    ,$CRLF              ;TYPE <CR><LF>
037426  004737  031250              10$:    JSR     PC,HLT000           ;CHECK IF HALT ON ERROR
037432  005237  001630                      INC     CNTCNT              ;INCREMENT CONTROLLER ERROR COUNT
037436  022737  000012  001630              CMP     #T.CONT,CNTCNT      ;CHECK IF THRESHOLD EXCEEDED
037444  103006                              BHIS    15$                 ;NO, CONTINUE
037446  104401  060533                      TYPE    ,ERR000             ;TYPE "FATAL ERROR"
037452  104401  061072                      TYPE    ,ERR008             ;TYPE "CONTROLLER ERROR THRESHOLD EXCEEDED"
037456  000000                              HALT
037460  000776                              BR      .-2
                                
037462  105765  000150              15$:    TSTB    P.DSTT(R5)          ;CHECK IF FIRST ERROR
037466  100005                              BPL     20$                 ;NO, REQUEUE COMMAND
037470  042765  100200  000150              BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR ERROR ENQUEUED AND FIRST ERROR
037476  005037  001566                      CLR     ERRPRO              ;CLEAR ERROR BEING PROCESSED
037502  004037  051406              20$:    JSR     R0,Q.PUSH           ;REQUEUE COMMAND IN COMMAND
037506  001324                              CINITQ                      ;  INITIATION QUEUE
037510  000200                              RTS     R0                  ;RETURN
```

```
                                        .SBTTL  TEST FOR DROP DRIVE ERRORS
                                        ;
037512  032765  000030  000036  TSTDRP: BIT     #ACLO!SPDLSS,P.DS(R5) ;CHECK IF AC LOW OR SPEED LOSS
037520  001016                          BNE     10$             ;YES REPORT ERROR
                                        ;       CHECK IF ANY OF THE FOLLOWING ERRORS ARE SET
                                        ;               UNSAFE
                                        ;               ILLEGAL DISK ADDRESS
                                        ;               WRITE LOCK ERROR
                                        ;               DRIVE TYPE ERROR
                                        ;               FORMAT ERROR
                                        ;               NON-EXECUTABLE DRIVE FUNCTION
037522  032765  046064  000034          BIT     #UNS!IDAE!WLE!DTYE!FMTE!NXF,P.ER(R5)
                                        ;
037530  001026                          BNE     12$             ;YES, REPORT ERROR
                                        ;       CHECK IF ANY OF THE FOLLOWING FLAGS ARE SET
                                        ;               DRIVE ERROR NOT REPORTED
                                        ;               DRIVE READY NOT SET AFTER START SPINDLE
                                        ;               VOLUME VALID NOT SET AFTER PACK ACKNOWLEDGE
037532  032765  006010  000212          BIT     #BIT3!BIT10!BIT11,P.ERR(R5)
                                        ;
037540  001072                          BNE     18$             ;YES, GO REPORT ERROR
                                        ;
                                        ;       CHECK IF ANY OF THE FOLLOWING FLAGS ARE SET
                                        ;               DRIVE HARD ERROR
                                        ;               DRIVE STATUS CHANGE DID NOT CLEAR
                                        ;               UNEXPECTED ATTN FROM DRIVE COMMAND
                                        ;               ATTENTION SET BUT DSC AND FAULT RESET
037542  032765  004070  000014          BIT     #DRVHRD!DRVDSC!UEXATT!NODSC,P.PRST(R5)
                                        ;
037550  001116                          BNE     32$             ;YES REPORT ERROR
037552  005720                          TST     (R0)+           ;GET NORMAL RETURN ADDRESS
037554  000200                          RTS     R0              ;RETURN
                                        ;
037556  004737  030000          10$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
037562  032765  000010  000036          BIT     #ACLO,P.DS(R5)  ;CHECK IF AC LOW
037570  001403                          BEQ     11$             ;NO, CONTINUE
037572  104401  062103                  TYPE    ,ERR033         ;TYPE "AC LOW"
037576  000472                          BR      30$             ;GO GET ERROR QUALIFICATION
                                        ;
037600  104401  062112          11$:    TYPE    ,ERR034         ;TYPE "SPEED LOSS"
037604  000467                          BR      30$             ;GET ERROR QUALIFICATION
                                        ;
037606  004737  030000          12$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
037612  032765  040000  000034          BIT     #UNS,P.ER(R5)   ;CHECK IF DRIVE UNSAFE
037620  001403                          BEQ     13$             ;NO, CONTINUE
037622  104401  062622                  TYPE    ,ERR056         ;TYPE "DRIVE UNSAFE"
037626  000456                          BR      30$             ;GET ERROR QUALIFICATION
                                        ;
037630  032765  000040  000034  13$:    BIT     #DTYE,P.ER(R5)  ;CHECK IF DRIVE TYPE ERROR
037636  001403                          BEQ     14$             ;NO, CONTINUE
037640  104401  062125                  TYPE    ,ERR035         ;TYPE "DRIVE TYPE ERROR"
037644  000447                          BR      30$             ;GET ERROR QUALIFIER BITS
                                        ;
037646  032765  000020  000034  14$:    BIT     #FMTE,P.ER(R5)  ;CHECK IF FORMAT ERROR
037654  001403                          BEQ     15$             ;NO, CONTINUE
037656  104401  062142                  TYPE    ,ERR036         ;TYPE FORMAT ERROR
```

```
7932  037662  000440                          BR      30$             ;GET ERROR QUAILIFER BITS
7933
7934  037664  032765  000004  000034  15$:    BIT     #NXF,P.ER(R5)   ;CHECK IF NON-EXECUTABLE DRIVE FUNCTION
7935  037672  001403                          BEQ     16$             ;NO, CONTINUE
7936  037674  104401  062155                  TYPE    ,ERR037         ;TYPE ILLEGAL DRIVE FUNCTION
7937  037700  000431                          BR      30$             ;GET ERROR QUALIFIER BITS
7938
7939  037702  032765  002000  000034  16$:    BIT     #IDAE,P.ER(R5)  ;CHECK IF ILLEGAL DISK ADDRESS
7940  037710  001403                          BEQ     17$             ;NO, CONTINUE
7941  037712  104401  062052                  TYPE    ,ERR031         ;TYPE "ILLEGAL DISK ADDRESS"
7942  037716  000422                          BR      30$             ;GET ERROR QUALIFER BITS
7943
7944  037720  104401  062066          17$:    TYPE    ,ERR032         ;TYPE "WRITE LOCK ERROR"
7945  037724  000417                          BR      30$             ;GET ERROR QUALIFIER BITS
7946
7947  037726  004737  030000          18$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
7948  037732  032765  002000  000212          BIT     #BIT10,P.ERR(R5) ;CHECK IF DRIVE NOT READY AFTER START SPINDLE
7949  037740  001403                          BEQ     19$             ;NO, CONTINUE
7950  037742  104401  062514                  TYPE    ,ERR053+1       ;PRINT "DRIVE NOT READY AFTER START SPINDLE"
7951  037746  000406                          BR      30$             ;GET ERROR QUALIFIER BITS
7952
7953  037750  032765  004000  000212  19$:    BIT     #BIT11,P.ERR(R5) ;CHECK IF VOLUME VALID DID NOT SET
7954                                                                  ;  AFTER PACK ACKNOWLEDGE
7955  037756  001404                          BEQ     31$             ;NO, CONTINUE
7956  037760  104401  062555                  TYPE    ,ERR054+1       ;PRINT "VOLUME VAID NOT SET AFTER
7957                                                                  ;  PACK ACKNOWLEDGE INDSTRUCTION"
7958  037764  104401  001165          30$:    TYPE    ,$CRLF          ;TYPE <CR><LF>
7959  037770  032765  000010  000212  31$:    BIT     #BIT3,P.ERR(R5) ;CHECK IF ERROR NOT INDICATED BY FAULT
7960  037776  001405                          BEQ     42$             ;NO, CHECK OTHER QUALIFIER BITS
7961  040000  104401  062635                  TYPE    ,ERR057         ;TYPE "ERROR NOT INDICATED BY FAULT"
7962  040004  000402                          BR      42$             ;CHECK OTHER QUALIFIER BITS
7963
7964  040006  004737  030000          32$:    JSR     PC,PRI000       ;PRINT ERROR HEADER
7965  040012  032765  000020  000014  42$:    BIT     #DRVHRD,P.PRST(R5) ;CHECK IF HARD DRIVE ERROR
7966  040020  001402                          BEQ     43$             ;NO, CHECK OTHER QUALIFIER BITS
7967  040022  104401  061723                  TYPE    ,ERR027         ;TYPE "DRIVE HARD ERROR"
7968  040026  032765  004000  000014  43$:    BIT     #NODSC,P.PRST(R5) ;CHECK IF ATTENTION BUT
7969                                                                  ;  NO DRIVE STATUS CHANGE OR FAULT
7970  040034  001402                          BEQ     44$             ;NO, CHECK OTHER QUALIFIER BITS
7971  040036  104401  062676                  TYPE    ,ERR058         ;TYPE "ATTN BUT NO DSC OR FAULT"
7972  040042  032765  000040  000014  44$:    BIT     #DRVDSC,P.PRST(R5) ;CHECK IF DRIVE STATUS CHANGE DID
7973                                                                  ;  NOT CLEAR
7974  040050  001402                          BEQ     45$             ;NO, CHECK OTHER ERROR QUALIFIER BITS
7975  040052  104401  061742                  TYPE    ,ERR028         ;TYPE "DRIVE STATUS CHANGE DID NOT CLEAR"
7976  040056  032765  000010  000014  45$:    BIT     #UEXATT,P.PRST(R5) ;CHECK IF UNEXPECTED ATTENTION FROM
7977                                                                  ;  DRIVE COMMAND
7978  040064  001402                          BEQ     46$             ;NO, CHECK OTHER QUALIFIER BITS
7979  040066  104401  062032                  TYPE    ,ERR030         ;TYPE "UNEXPECTED ATTN FROM DRIVE COMMAND"
7980  040072  005265  000206          46$:    INC     P.NER(R5)       ;INCREMENT OTHER ERROR COUNT
7981  040076  032777  020000  141034          BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
7982  040104  001015                          BNE     60$             ;YES, CHECK IF HALT ON ERROR
7983  040106  005765  000150                  TST     P.DSTT(R5)      ;CHECK IF ERROR DURING ERROR RECOVERY
7984  040112  100402                          BMI     51$             ;NO, PRINT STATUS
7985  040114  104401  063721                  TYPE    ,ERR202         ;TYPE "ERROR DURING ERROR RECOVERY"
7986  040120  004737  026074          51$:    JSR     PC,PRISTT       ;PRINT CONTROLLER REGISTERS
7987  040124  004737  025720                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
```

```
 7988  040130  004737  030244                 JSR     PC,PRI002       ;PRINT CURRENT COMMAND
 7989  040134  004737  030066                 JSR     PC,PRI001       ;PRINT PREVIOUS COMMAND
 7990  040140  004737  031250         60$:    JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
 7991  040144  032777  000100  140766         BIT     #SW6,@SWR       ;CHECK FOR SWITCH 6
 7992  040152  001431                         BEQ     63$             ;NO DROP DRIVE
 7993  040154  032765  000030  000036         BIT     #ACLO!SPDLSS,P.DS(R5) ;CHECK IF UNSAFE, ACLOW AND SPEED LOSS
 7994  040162  001004                         BNE     61$
 7995  040164  032765  040000  000034         BIT     #UNS,P.ER(R5)
 7996  040172  001421                         BEQ     63$
 7997  040174  105765  000120         61$:    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
 7998  040200  001012                         BNE     62$
 7999  040202  105265  000146                 INCB    P.RECT(R5)      ;INCREMENT ERROR RETRY COUNT
 8000  040206  122765  000010  000146         CMPB    #8.,P.RECT(R5)  ;CHECK IF RETRY THRESHOLD EXCEEDED
 8001  040214  103404                         BLO     62$             ;YES, INDICATE UNSUCCESSFUL RECOVERY
 8002  040216  004037  051406                 JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 8003  040222  001324                         CINITQ                  ;  INITIATION QUEUE
 8004  040224  000402                         BR      57$             ;RETURN
 8005
 8006  040226  004737  031502         62$:    JSR     PC,UNRECY       ;INDICATE UNSUCCESSFUL RECOVERY
 8007  040232  011000         57$:    MOV     (R0),R0
 8008  040234  000200                         RTS     R0
 8009
 8010  040236  004737  022642         63$:    JSR     PC,BUFREL       ;RELEASE BUFFER
 8011  040242  004737  013004                 JSR     PC,DROP         ;DROP DRIVE
 8012  040246  004737  022342                 JSR     PC,GETBUF       ;GET NEW BUFFER
 8013  040252  005037  001566                 CLR     ERRPRO          ;CLEAR ERROR BEING PROCESSED
 8014  040256  011000                         MOV     (R0),R0         ;LOAD ERROR RETURN
 8015  040260  000200                         RTS     R0              ;RETURN
```

```
                                        .SBTTL  ERROR RECOVERY SEQUENCE NORMAL RETURN

040262  032765  004000  000150  ERCVY1: BIT     #BIT11,P.DSTT(R5) ;CHECK IF RECALIBRATE ISSUED
040270  001415                          BEQ     EC3$              ;NO, CONTINUE
040272  042765  004000  000150          BIC     #BIT11,P.DSTT(R5) ;CLEAR RECALIBRATE ISSUED
040300  052765  010000  000150  15$:     BIS     #BIT12,P.DSTT(R5) ;SET SEEK ISSUED
040306  112765  000117  000001          MOVB    #SEEK,P.CMND(R5) ;ISSUE SEEK INSTRUCTION
040314  004037  051406                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
040320  001324                          CINITQ                    ;  INITIATION QUEUE
040322  000207                          RTS     PC                ;RETURN

040324  032765  010000  000150  EC3$:   BIT     #BIT12,P.DSTT(R5) ;CHECK IF SEEK ISSUED
040332  001427                          BEQ     10$               ;NO, CONTINUE
040334  042765  010000  000150          BIC     #BIT12,P.DSTT(R5) ;CLEAR SEEK ISSUED
040342  105765  000211                  TSTB    P.SEEK(R5)        ;CHECK IF SEEK TO PREVIOUS POSITION
040346  001403                          BEQ     5$                ;NO, PROCESS DATA TRANSFER
040350  004737  036276                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
040354  000207                          RTS     PC                ;RETURN

040356  122765  000021  000147  5$:     CMPB    #21,P.RERD(R5)    ;CHECK IF OFFSET
040364  101053                          BHI     20$               ;NO, REISSUE COMMAND
040366  052765  020000  000150          BIS     #BIT13,P.DSTT(R5) ;SET OFFSET ISSUED
040374  112765  000115  000001          MOVB    #OFFSET,P.CMND(R5) ;ISSUE OFFSET
040402  004037  051406                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
040406  001324                          CINITQ                    ;  INITIATION QUEUE
040410  000207                          RTS     PC                ;RETURN

040412  032765  020000  000150  10$:    BIT     #BIT13,P.DSTT(R5) ;CHECK IF DO PREVIOUS OFFSET
040420  001453                          BEQ     EC4$              ;NO, CONTINUE
040422  042765  020000  000150          BIC     #BIT13,P.DSTT(R5) ;CLEAR OFFSET ISSUED
040430  122765  000021  000147          CMPB    #21,P.RERD(R5)    ;CHECK IF OFFSET +400 MICRO-INCHES
040436  001424                          BEQ     15$               ;YES, GO ISSUE READ
040440  122765  000024  000147          CMPB    #24,P.RERD(R5)    ;CHECK IF OFFSET -400 MICRO-INCHES
040446  001420                          BEQ     15$               ;YES, GO ISSUE READ
040450  122765  000027  000147          CMPB    #27,P.RERD(R5)    ;CHECK IF OFFSET +800 MICRO-INCHES
040456  001414                          BEQ     15$               ;YES, GO ISSUE READ
040460  122765  000032  000147          CMPB    #32,P.RERD(R5)    ;CHECK IF OFFSET -800 MICRO-INCHES
040466  001410                          BEQ     15$               ;YES, GO ISSUE READ
040470  122765  000035  000147          CMPB    #35,P.RERD(R5)    ;CHECK IF OFFSET +1200 MICRO-INCHES
040476  001404                          BEQ     15$               ;YES, GO ISSUE READ
040500  122765  000040  000147          CMPB    #40,P.RERD(R5)    ;CHECK IF OFFSET -1200 MICRO-INCHES
040506  001002                          BNE     20$               ;NO, DO NOT INCREMENT REREAD COUNT
040510  105265  000147          15$:    INCB    P.RERD(R5)        ;INCREMENT REREAD COUNT
040514  116565  000123  000001  20$:    MOVB    P.RCMD(R5),P.CMND(R5) ;LOAD COMMAND
040522  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
040530  001003                          BNE     25$               ;NO, ISSUE COMMAND
040532  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;YES, ISSUE WRITE BEFOR WRITE CHECK
040540  004037  051406          25$:    JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN COMMAND
040544  001324                          CINITQ                    ;  INITIATION QUEUE
040546  000207                          RTS     PC                ;RETURN

040550  032765  000100  000150  EC4$:   BIT     #BIT6,P.DSTT(R5)  ;CHECK IF WAITING FOR READ STATUS
                                                                  ;  FOR POSITIONING TIME OUT
040556  001436                          BEQ     EC5$              ;NO, CONTINUE
040560  042765  000100  000150          BIC     #BIT6,P.DSTT(R5)  ;CLEAR WAITING FOR READ STATUS
040566  113765  001666  000001          MOVB    POSCMD,P.CMND(R5) ;RESTORE COMMAND OF ERROR
```

```
8072  040574  032765  000200  000036          BIT     #DRDY,P.DS(R5)  ;CHECK IF DRIVE READY
8073  040602  001004                          BNE     10$             ;YES, REISSUE COMMAND
8074  040604  004037  030464                  JSR     R0,PRI100       ;PRINT DRIVE TIME OUT DURING POSITIONING
8075  040610  062435                          ERR052                  ;  AND DROP DRIVE
8076  040612  000207                  5$:     RTS     PC              ;RETURN
8077
8078  040614  004037  030654          10$:    JSR     R0,PRI200       ;PRINT DRIVE TIME OUT DURING
8079  040620  062435                          ERR052                  ;  POSITIONING
8080  040622  004037  031300                  JSR     R0,NERSTT       ;TAKE OTHER ERROR STATICS
8081  040626  040612                          5$                      ;NO RECOVERY RETURN
8082  040630  052765  004000  000150          BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
8083  040636  112765  000113  000001          MOVB    #RECAL,P.CMND(R5) ;ISSUE RECALIBRATE
8084  040644  004037  051406                  JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
8085  040650  001324                          CINITQ                  ;  INITIATION QUEUE
8086  040652  000207                          RTS     PC              ;RETURN
8087
8088  040654  032765  000020  000150  EC5$:   BIT     #BIT4,P.DSTT(R5) ;CHECK IF HEADER ERROR
8089  040662  001002                          BNE     1$              ;YES, GO PROCESS ERROR
8090  040664  000137  041400                  JMP     EC6$            ;NO, CONTINUE
8091
8092  040670  042765  000020  000150  1$:     BIC     #BIT4,P.DSTT(R5) ;CLEAR READ ALL HEADERS ISSUED
8093  040676  010346                          MOV     R3,-(SP)        ;STORE R3 ON THE STACK
8094  040700  016546  000056                  MOV     P.B11(R5),-(SP) ;STORE SECTOR READ
8095  040704  042716  177017                  BIC     #^C<M.SECT>,(SP) ;KEEP ONLY SECTOR COUNT
8096  040710  006216                          ASR     (SP)            ;SHIFT 4 BITS RIGHT
8097  040712  006216                          ASR     (SP)
8098  040714  006216                          ASR     (SP)
8099  040716  006216                          ASR     (SP)            ;GET SECTOR COUNT
8100  040720  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
8101  040722  116516  000004                  MOVB    P.SECT(R5),(SP) ;GET DESIRED SECTOR
8102  040726  162616                          SUB     (SP)+,(SP)      ;CALCULATE DIFFERENCE
8103  040730  100410                          BMI     2$              ;BRANCH IF IN LOWER PART OF BUFFER
8104  040732  012703  066204                  MOV     #BUFADD-6,R3    ;LOAD BASE OF HIGHER PART OF BUFFER
8105  040736  006316                          ASL     (SP)            ;MULTIPLY SECTOR COUNT BY 6
8106  040740  011646                          MOV     (SP),-(SP)
8107  040742  006316                          ASL     (SP)
8108  040744  062616                          ADD     (SP)+,(SP)
8109  040746  162603                          SUB     (SP)+,R3        ;CALCULATE HEADER ADDRESS
8110  040750  000407                          BR      3$              ;CHECK HEADER
8111
8112  040752  005416                  2$:     NEG     (SP)            ;MAKE COUNT POSITIVE
8113  040754  006316                          ASL     (SP)            ;MULTIPLY DIFFERENCE BY 6
8114  040756  011603                          MOV     (SP),R3
8115  040760  006316                          ASL     (SP)
8116  040762  062603                          ADD     (SP)+,R3
8117  040764  062703  066000                  ADD     #HDBUFF-6,R3    ;CALCULATE HEADER ADDRESS
8118  040770  012337  001646          3$:     MOV     (R3)+,HEAD1     ;GET FIRST WORD OF HEADER
8119  040774  012337  001650                  MOV     (R3)+,HEAD2     ;GET SECOND WORD OF HEADER
8120  041000  011337  001652                  MOV     (R3),HEAD3      ;GET THIRD WORD OF HEADER
8121  041004  012603                          MOV     (SP)+,R3        ;RESTORE R3
8122  041006  004737  036776                  JSR     PC,RS↑AT        ;RESTORE STATUS
8123  041012  013746  001646                  MOV     HEAD1,-(SP)     ;STORE FIRST WORD OF THE HEADER
8124  041016  043716  001650                  BIC     HEAD2,(SP)      ;HD1 & ↑HD2
8125  041022  013746  001650                  MOV     HEAD2,-(SP)     ;STORE SECOND WORD OF HEADER
8126  041026  043716  001646                  BIC     HEAD1,(SP)      ;↑HD1 & HD2
8127  041032  052616                          BIS     (SP)+,(SP)      ;GENERATE EXPECTED HEADER VRC
```

```
 8128  041034  023726* 001652                    CMP     HEAD3,(SP)+     ;CHECK IF HEADER VRC IS CORRECT
 8129  041040  001501                            BEQ     15$             ;NO, CONTINUE
 8130  041042  004037  030654                    JSR     R0,PRI200       ;PRINT HEADER VRC ERROR
 8131  041046  062262                            ERR043
 8132  041050  105765  000120                    TSTB    P.IERC(R5)      ;CHECK IF INHIBIT ERROR RECOVERY
 8133  041054  001016                            BNE     5$              ;YES, INDICATE UNSUCCESSFUL RECOVERY
 8134  041056  105265  000147            4$:     INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
 8135  041062  122765  000043  000147            CMPB    #43,P.RERD(R5)  ;CHECK IF UNRECOVERABLE
 8136  041070  101410                            BLOS    5$              ;YES, INDICATE UNRECOVERABLE ERROR
 8137  041072  122765  000121  000123            CMPB    #RDDATA,P.RCMD(R5) ;CHECK IF READ DATA COMMAND
 8138  041100  001407                            BEQ     6$              ;YES, START ERROR RECOVERY
 8139  041102  122765  000021  000147            CMPB    #21,P.RERD(R5)  ;CHECK IF UNRECOVERABLE WRITE
 8140  041110  101021                            BHI     8$              ;NO, START ERROR RECOVERY
 8141  041112  004737  031502            5$:     JSR     PC,UNRECY       ;INDICATE UNRECOVERABLE ERROR
 8142  041116  000207                            RTS     PC              ;RETURN
 8143
 8144  041120  122765  000021  000147    6$:     CMPB    #21,P.RERD(R5)  ;CHECK IF BEGINNING OFFSETTING OPERATION
 8145  041126  001007                            BNE     7$              ;NO, DETERMINE PROPER OFFSET
 8146  041130  010446                            MOV     R4,-(SP)        ;STORE R4 ON THE STACK
 8147  041132  116504  000000                    MOVB    P.DRVN(R5),R4   ;GET DRIVE NUMBER FOR INDEX
 8148  041136  146437  001514  001513            BICB    INTMSK(R4),O.OVER ;DO NOT DO IMPLIED SEEKS
 8149  041144  012604                            MOV     (SP)+,R4        ;RESTORE R4
 8150  041146  004037  037026            7$:     JSR     R0,DETOFF       ;DETERMINE OFFSET VALUE
 8151  041152  041242                            12$                     ;OFFSET ISSUED RETURN
 8152  041154  116565  000123  000001    8$:     MOVB    P.RCMD(R5),P.CMND(R5) ;STORE COMMAND
 8153  041162  013765  001642  000002            MOV     RTYCYL,P.CYLN(R5) ;GET CYLINDER
 8154  041170  013765  001640  000004            MOV     RTYSEC,P.SECT(R5) ;GET TRACK AND SECTOR
 8155  041176  013765  001634  000010            MOV     RTYBA,P.BALO(R5) ;GET BUS ADDRESS
 8156  041204  013765  001636  000012            MOV     RTYWC,P.WC(R5) ;GET WORD COUNT
 8157  041212  005465  000012                    NEG     P.WC(R5)        ;MAKE IT POSITIVE
 8158  041216  122765  000131  000001            CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK COMMAND
 8159  041224  001003                            BNE     10$             ;NO, GO ISSUE COMMAND
 8160  041226  052765  000200  000014            BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
 8161  041234  004037  051406            10$:    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK ON COMMAND
 8162  041240  001324                            CINITQ                  ;  INITIATION QUEUE
 8163  041242  000207                    12$:    RTS     PC              ;RETURN
 8164
 8165  041244  023765  001646  000030    15$:    CMP     HEAD1,P.DCYL(R5) ;CHECK IF CYLINDER CORRECT
 8166  041252  001420                            BEQ     20$             ;YES, CHECK SECTOR TRACK AND FORMAT
 8167  041254  004037  030654                    JSR     R0,PRI200       ;PRINT MISPOSITIONING
 8168  041260  063075                            ERR064
 8169  041262  004037  031346                    JSR     R0,OPISTT       ;GO TAKE STATISTICS
 8170  041266  041312                            17$                     ;UNRECOVERABLE ERROR RETURN
 8171  041270  052765  004000  000150            BIS     #BIT11,P.DSTT(R5) ;SET RECALIBRATE ISSUED
 8172  041276  112765  000113  000001            MOVB    #RECAL,P.CMND(R5) ;ISSUE RECALIBRATE
 8173  041304  004037  051406                    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN
 8174  041310  001324                            CINITQ                  ;  COMMAND INITIATION QUEUE
 8175  041312  000207                    17$:    RTS     PC              ;RETURN
 8176
 8177  041314  005046                    20$:    CLR     -(SP)           ;MAKE ROOM ON STACK
 8178  041316  113766  001641  000001            MOVB    RTYTRK,1(SP)    ;GET TRACK
 8179  041324  006216                            ASR     (SP)            ;SHIFT RIGHT 3 BITS
 8180  041326  006216                            ASR     (SP)
 8181  041330  006216                            ASR     (SP)
 8182  041332  153716  001640                    BISB    RTYSEC,(SP)     ;GENERATE EXPECTED SECOND WORD
 8183  041336  052716  140000                    BIS     #140000,(SP)    ;PUT IN GO TRACK BITS
```

```
 8184  041342  022637  001650                    CMP     (SP)+,HEAD2     ;CHECK IF SECOND WORD IS CORRECT
 8185  041346  001407                            BEQ     25$             ;CHECK FOR BAD SECTOR
 8186  041350  004037  030654                    JSR     R0,PRI200       ;INDICATE OPERATION INCOMPLETE
 8187  041354  062243                            ERR042
 8188  041356  004037  031346                    JSR     R0,OPISTT       ;GO TAKE STATISTICS
 8189  041362  041312                            17$                     ;UNRECOVERABLE RETURN
 8190  041364  000673                            BR      8$              ;GO ISSUE LAST COMMAND
 8191
 8192  041366  004037  030654          25$:      JSR     R0,PRI200       ;TYPE HEADER ERROR
 8193  041372  064232                            ERR209
 8194  041374  000137  041056                    JMP     4$              ;GO TRY TO RETRY COMMAND
 8195
 8196  041400  122765  000021  000147  EC6$:     CMPB    #21,P.RERD(R5)  ;CHECK IF REISSUE DATA TRANSFER
 8197  041406  101036                            BHI     10$             ;YES, INDICATE SUCESSFUL RECOVERY
 8198  041410  001424                            BEQ     5$              ;NO, OFFSET = +400 MICRO-INCHES
 8199                                                                    ;   ISSUE DATA TRANSFER
 8200  041412  122765  000024  000147            CMPB    #24,P.RERD(R5)  ;CHECK IF OFFSET -400 MICRO-INCHES
 8201  041420  001420                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
 8202  041422  122765  000027  000147            CMPB    #27,P.RERD(R5)  ;CHECK IF OFFSET +800 MICRO-INCHES
 8203  041430  001414                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
 8204  041432  122765  000032  000147            CMPB    #32,P.RERD(R5)  ;CHECK IF OFFSET -800 MICRO-INCHES
 8205  041440  001410                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
 8206  041442  122765  000035  000147            CMPB    #35,P.RERD(R5)  ;CHECK IF OFFSET +1200 MICRO-INCHES
 8207  041450  001404                            BEQ     5$              ;YES, ISSUE DATA TRANSFER
 8208  041452  122765  000040  000147            CMPB    #40,P.RERD(R5)  ;CHECK IF OFFSET -1200 MICRO-INCHES
 8209  041460  001011                            BNE     10$             ;NO, INDICATE SUCESSFUL RECOVERY
 8210  041462  105265  000147          5$:       INCB    P.RERD(R5)      ;INCREMENT REREAD COUNT
 8211  041466  116565  000123  000001            MOVB    P.RCMD(R5),P.CMND(R5) ;REISSUE COMMAND
 8212  041474  004037  051406                    JSR     R0,Q.PUSH       ;ENQUEUE PARAMETER BLOCK IN COMMAND
 8213  041500  001324                            CINITQ                  ;   INITIATION QUEUE
 8214  041502  000207                            RTS     PC              ;RETURN
 8215
 8216  041504  032765  040000  000150  10$:      BIT     #BIT14,P.DSTT(R5) ;CHECK IF SERVICING BAD SECTOR
 8217  041512  001403                            BEQ     11$             ;NO, CONTINUE
 8218  041514  005765  000146                    TST     P.RECT(R5)      ;CHECK IF IN RETRY SEQUENCE
 8219  041520  001472                            BEQ     EC7$            ;NO, GO TO BAD SECTOR SERVICE ROUTINE
 8220  041522  105765  000210          11$:      TSTB    P.ASSN(R5)      ;CHECK IF DRIVE IS BEING ASSIGNED
 8221  041526  001403                            BEQ     12$             ;NO, CHECK IF SEEK TO PREVIOUS CYLINDER
 8222  041530  004737  036276                    JSR     PC,SURECY       ;INIDICATE SUCCESSFUL RECOVERY
 8223  041534  000207                            RTS     PC              ;RETURN
 8224
 8225  041536  122765  000117  000001  12$:      CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK TO PREVIOUS CYLINDER
 8226  041544  001404                            BEQ     14$             ;YES, INDICATE SUCCESSFUL RECOVERY
 8227  041546  122765  000140  000001            CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE
 8228  041554  001003                            BNE     16$             ;NO, PROCESS DATA TRANSFER
 8229  041556  004737  036276          14$:      JSR     PC,SURECY       ;INDICATE SUCCESSFUL RECOVERY
 8230  041562  000207                            RTS     PC              ;RETURN
 8231
 8232  041564  004037  025032          16$:      JSR     R0,CHKADD       ;CHECK SECTOR, TRACK, CYLINDER, BUS ADDRESS
 8233                                                                    ;   AND WORD COUND
 8234  041570  041626                            20$                     ;ERROR RETURN
 8235  041572  005037  177776                    CLR     PS              ;ALLOW RK06 INTERRUPTS
 8236  041576  122765  000121  000001            CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ DATA
 8237  041604  001021                            BNE     30$             ;NO, CHECKJ IF ;WRITE CHECK
 8238  041606  004037  023304                    JSR     R0,BUFER1       ;FIND FIRST DATA ERROR
 8239  041612  041630                            25$                     ;ERROR RETURN
```

```
8240  041614  013737  001452  177776  18$:    MOV     RKPRI,PS          ;LOCK OUT RK06 INTERRUPTS
8241  041622  004737  036276                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
8242  041626  000207                  20$:    RTS     PC                ;RETURN
8243
8244  041630  013737  001452  177776  25$:    MOV     RKPRI,PS          ;LOCK OUT RK06 INTERRUPTS
8245  041636  052765  000004  000212          BIS     #BIT2,P.ERR(R5)   ;INDICATE DATA COMPARE ERROR
8246  041644  000137  035310                  JMP     E10$              ;PROCESS DATA COMPARE ERROR
8247
8248  041650  122765  000131  000001  30$:    CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK
8249  041656  001356                          BNE     18$               ;NO, INDICATE SUCCESSFUL RECOVERY
8250  041660  032765  000200  000014          BIT     #W.WCK,P.PRST(R5) ;CHECK IF WRITE CHECK ISSUED
8251  041666  001752                          BEQ     18$               ;YES, INDICATE SUCCESSFUL RECOVERY
8252  041670  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;CLEAR WRITE BEFORE WRITE CHECK
8253  041676  004037  051406                  JSR     R0,Q.PUSH         ;ENQUEUE COMMAND IN COMMAND INITIATION QUEUE
8254  041702  001324                          CINITQ
8255  041704  000207                          RTS     PC                ;RETURN
8256
8257  041706  004037  025032          EC7$:   JSR     R0,CHKADD         ;CHECK IF BUS ADD, WORD COUNT, SECTOR,
8258                                                                    ;  TRACK AND CYLINDER ARE CORRECT
8259  041712  042000                          9$                        ;ERROR RETURN
8260  041714  042765  040000  000150          BIC     #BIT14,P.DSTT(R5) ;CLEAR SERVICING BAD SECTOR
8261  041722  004737  042142                  JSR     PC,BSTRAN         ;CALCULATE NUMBER OF WORDS TRANSFERRED
8262  041726  013765  001660  000012          MOV     BSWC,P.WC(R5)     ;LOAD BUS ADDRESS FOR NEXT DATA TRANSFER
8263  041734  100022                          BPL     10$               ;CHECK IF DATA TRANSFER FINISHED
8264  041736  013765  001654  000004          MOV     BSSECT,P.SECT(R5) ;LOAD SECTOR AND TRACK FOR NEXT COMMAND
8265  041744  013765  001656  000002          MOV     BSCYLN,P.CYLN(R5) ;LOAD CYLINDER FOR NEXT COMMAND
8266  041752  013765  001662  000010          MOV     BSBA,P.BALO(R5)   ;LOAD BUS ADDRESS FOR NEXT DATA TRANSFER
8267  041760  052765  000200  000014          BIS     #W.WCK,P.PRST(R5) ;GO ISSUE WRITE
8268  041766  004037  051406          5$:     JSR     R0,Q.PUSH         ;ENQUEUE COMMAND IN COMMAND INITIATION
8269  041772  001324                          CINITQ                    ;  QUEUE
8270  041774  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR PROCESSING FLAG
8271  042000  000207                  9$:     RTS     PC                ;RETURN
8272
8273  042002  105765  000210          10$:    TSTB    P.ASSN(R5)        ;CHECK IF PACK IS BEING WRITTEN
8274  042006  001012                          BNE     13$               ;YES, CHECK IF SEEK TO NEXT CYLINDER
8275  042010  004737  022642                  JSR     PC,BUFREL         ;RELEASE BUFFER
8276  042014  004037  051406                  JSR     R0,Q.PUSH         ;ENQUEUE PARMETER BLOCK IN DRIVE
8277  042020  001314                          AVAILQ                    ;  AVAILIABLE QUEUE
8278  042022  004737  022342                  JSR     PC,GETBUF         ;GET NEXT BUFFER FOR DAT TRANSFER
8279  042026  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSING
8280  042032  000207                          RTS     PC                ;RETURN
8281
8282  042034  005037  001566          13$:    CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSED
8283  042040  013765  001654  000026          MOV     BSSECT,P.DTS(R5)  ;LOAD NEXT SECTOR AND TRACK
8284  042046  013765  001656  000030          MOV     BSCYLN,P.DCYL(R5) ;LOAD NEXT CYLINDER
8285  042054  032777  040000  137056          BIT     #SW14,@SWR        ;CHECK IF LOOP ON CURRENT OPERATION
8286  042062  001002                          BNE     14$               ;YES, ISSUE SEEK TO PREVIOUS CYLINDER
8287  042064  000137  015276                  JMP     WV0$              ;NO, CONTINUE TO WRITE PACK
8288
8289  042070  004737  022642          14$:    JSR     PC,BUFREL         ;RELEASE BUFFER
8290  042074  032765  002000  000014          BIT     #DRPDRV,P.PRST(R5) ;CHECK IF DROP DRIVE
8291  042102  001402                          BEQ     15$               ;NO ISSUE SEEK TO PREVIOUS CYLINDER
8292  042104  000137  016040                  JMP     WV1$              ;DROP DRIVE
8293
8294  042110  112765  177777  000211  15$:    MOVB    #-1,P.SEEK(R5)    ;SET SEEK TO PREVIOUS CYLINDER
8295  042116  016565  000136  000002          MOV     P.LCYL(R5),P.CYLN(R5) ;GET LAST CYLINDER
```

```
 8296  042124  016565  000140  000004          MOV     P.LSEC(R5),P.SECT(R5) ;GET LAST TRACK AND SECTOR
 8297  042132  112765  000117  000001          MOVB    #SEEK,P.CMND(R5) ;LOAD COMMAND
 8298  042140  000712                          BR      5$                ;GO ISSUE SEEK COMMAND
```

```
                                        .SBTTL  CALCULATE WORDS TRANSFERRED UNTIL BAD SECTOR

042142  016537  000026  001654  BSTRAN: MOV     P.DTS(R5),BSSECT ;STORE ADDRESS OF BAD SECTOR
042150  016537  000030  001656          MOV     P.DCYL(R5),BSCYLN
042156  105237  001654                  INCB    BSSECT          ;ADVANCE TO NEXT SECTOR ON PACK
042162  122737  000025  001654          CMPB    #21.,BSSECT     ;CHECK FOR SECTOR OVERFLOW
042170  103014                          BHIS    5$              ;NO, CONTINUE
042172  105037  001654                  CLRB    BSSECT          ;CLEAR SECTOR
042176  105237  001655                  INCB    BSTRCK          ;INCREMENT TRACK
042202  122737  000002  001655          CMPB    #2,BSTRCK       ;CHECK FOR TRACK OVERFLOW
042210  103004                          BHIS    5$              ;NO, CONTINUE
042212  105037  001655                  CLRB    BSTRCK          ;CLEAR TRACK
042216  005237  001656                  INC     BSCYLN          ;INCREMENT CYLINDER
042222  013746  001656          5$:     MOV     BSCYLN,-(SP)    ;STORE CYLINDER ON STACK
042226  166516  000002                  SUB     P.CYLN(R5),(SP) ;SUBTRACT STARTING CYLINDER
042232  011646                          MOV     (SP),-(SP)      ;SAVE DIFFERENCE
042234  006316                          ASL     (SP)            ;MULTIPLY CYLINDER BY 3
042236  062616                          ADD     (SP)+,(SP)
042240  005046                          CLR     -(SP)           ;MAKE ROOM ON STACK
042242  113716  001655                  MOVB    BSTRCK,(SP)     ;PLACE TRACK ON STACK
042246  062616                          ADD     (SP)+,(SP)      ;ADD TRACK
042250  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
042252  116516  000005                  MOVB    P.TRCK(R5),(SP) ;STORE TRACK
042256  162616                          SUB     (SP)+,(SP)      ;SUBTRACT INITIAL TRACK
042260  012746  000026                  MOV     #22.,-(SP)      ;STORE 22 ON STACK FOR MULTIPLICATION
042264  004737  054714                  JSR     PC,$MULT        ;MULTIPLY BY 22
042270  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
042272  113716  001654                  MOVB    BSSECT,(SP)     ;STORE BAD SECTOR
042276  062616                          ADD     (SP)+,(SP)      ;ADD BAD SECTOR ADDRESS
042300  005046                          CLR     -(SP)           ;MAKE ROOM ON THE STACK
042302  116516  000004                  MOVB    P.SECT(R5),(SP) ;STORE ORGINAL SECTOR ADDRESS
042306  162616                          SUB     (SP)+,(SP)      ;SUBTRACT ORGINAL SECTOR ADDRESS
042310  111666  000001                  MOVB    (SP),1(SP)      ;KEEP NUMBER OF SECTORS TRANSFERRED
042314  105016                          CLRB    (SP)
042316  011637  001664                  MOV     (SP),BSWORD     ;LOAD WORD COUNT
042322  006316                          ASL     (SP)            ;MULTIPLY BY 2
042324  066516  000010                  ADD     P.BALO(R5),(SP) ;GET STARTING ADDRESS FOR NEW DATA
042330  011637  001662                  MOV     (SP),BSBA       ;GET NEW BUS ADDRESS
042334  166516  000132                  SUB     P.RBAL(R5),(SP) ;GET NUMBER OF BYTES TRANSFERRED
042340  006216                          ASR     (SP)            ;CONVERT TO WORDS (DIVIDE BY 2)
042342  066516  000130                  ADD     P.RWC(R5),(SP)  ;DETERMINE REMAINING WORD COUNT
042346  012637  001660                  MOV     (SP)+,BSWC      ;STORE CALCULATED WORD COUNT
042352  005726                          TST     (SP)+           ;THROW AWAY MOST SIGNIFICANT BITS
042354  000207                  RTS     PC                      ;RETURN
```

```
                                        .SBTTL  BAD SECTOR HANDLING
                                
042356  032765  040000  000150  BDSECT: BIT     #BIT14,P.DSTT(R5) ;CHECK IF PRESENTLY SERVICING BAD SECTOR
042364  001401                  BEQ     5$                ;NO, CONTINUE
042366  000000                  HALT
042370  132765  000010  000210  5$:     BITB    #BIT3,P.ASSN(R5) ;CHECK IF READING PACK SERIAL NUMBER
042376  001157                  BNE     40$               ;YES, DETERMINE NEXT SECTOR
042400  004737  042142          JSR     PC,BSTRAN         ;CALCULATE WORDS TRANSFERRED
042404  122765  000121  000001  CMPB    #RDDATA,P.CMND(R5) ;CHECK IF READ COMMAND
042412  001076                  BNE     25$               ;NO, DO NO CHECK BUS ADDRESS AND WORD COUNT
042414  013746  001662          MOV     BSBA,-(SP)        ;STORE BUS ADDRESS
042420  162716  001000          SUB     #256.*2,(SP)      ;SUBTRACT ADDITIONAL SECTOR
042424  026526  000024          CMP     P.BAR(R5),(SP)+   ;CHECK IF BUS ADDRESS CORRECT
042430  001404                  BEQ     10$               ;YES, CONTINUE
042432  004037  037322          JSR     R0,CONERR         ;PRINT "BUS ADDRESS INVALID"
042436  063214                  ERR071
042440  000207                  RTS     PC                ;RETURN
                                
042442  016546  000022  10$:     MOV     P.WCR(R5),-(SP)   ;STORE WORD COUNT FOR COMPARISON
042446  062716  000400          ADD     #256.,(SP)        ;SUBTRACT OFF BAD SECTOR
042452  023726  001660          CMP     BSWC,(SP)+        ;CHECK IF WORD COUNT CORRECT
042456  001404                  BEQ     15$               ;YES, CONTINUE
042460  004037  037322          JSR     R0,CONERR         ;PRINT "WORD COUNT INVALID"
042464  063171                  ERR070
042466  000207                  RTS     PC                ;RETURN
                                
042470  042765  100200  000150  15$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
042476  022737  000400  001664          CMP     #256.,BSWORD      ;CHECK IF NO WORDS TRANSFERRED
042504  001413                  BEQ     20$               ;YES, DO NOT CHECK DATA
042506  004037  023304          JSR     R0,BUFER1         ;CHECK DATA READ
042512  043046                  50$                       ;ERROR RETURN
042514  063765  001664  000012          ADD     BSWORD,P.WC(R5) ;CALCULATE NEW WORD COUNT
042522  100410                  BMI     22$               ;CHECK IF DATA TRANSFER NOT COMPLETE
042524  000464                  BR      30$               ;DATA TRANSFER COMPLETE
                                
042526  052765  000200  000014  19$:    BIS     #W.WCK,P.PRST(R5) ;ISSUE WRITE BEFORE WRITE CHECK
042534  013765  001660  000012  20$:    MOV     BSWC,P.WC(R5)     ;GET NUMBER OF WORDS TO BE TRANSFERRED
042542  100055                  BPL     30$               ;CHECK IF DATA TRANSFER COMPLETE
042544  013765  001656  000002  22$:    MOV     BSCYLN,P.CYLN(R5) ;GET NEW CYLINDER
042552  013765  001654  000004          MOV     BSSECT,P.SECT(R5) ;GET NEW SECTOR
042560  013765  001662  000010          MOV     BSBA,P.BALO(R5)   ;GET NEW BUS ADDRESS
042566  004037  051406          JSR     R0,Q.PUSH         ;PUT PARAMETER IN COMMAND
042572  001324                  CINITQ                    ;   INITIATION QUEUE
042574  005765  000146          TST     P.RECT(R5)        ;CHECK IF ERROR RECOVERY IN PROGRESS
042600  001002                  BNE     23$               ;YES, DO NOT RESET ERROR PROCESSING IN PROGRESS
042602  005037  001566          CLR     ERRPRO            ;RESET ERROR PROCESSING IN PROGRESS
042606  000207                  RTS     PC        23$:    ;RETURN
                                
042610  042765  100200  000150  25$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
042616  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF WRITE CHECK COMMAND
042624  001343                  BNE     20$               ;NO, CONTINUE
042626  022737  000400  001664          CMP     #256.,BSWORD      ;CHECK IF NO WORDS TRANSFERRED
042634  001734                  BEQ     19$               ;YES, ISSUE NEXT DATA TRANSFER
042636  012746  000400          MOV     #256.,-(SP)       ;STORE EXCESS WORD COUNT ON STACK
042642  163716  001664          SUB     BSWORD,(SP)       ;CALCULATE NEW WORD COUNT
042646  012665  000012          MOV     (SP)+,P.WC(R5)    ;STORE NEW WORD COUNT
```

```
8399  042652  052765  040000  000150          BIS     #BIT14,P.DSTT(R5) ;INDICATE THAT BAD
8400                                                    ;  SECTOR IS BEING PROCESSED
8401  042660  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;DO WRITE CHECK PART OF COMMAND
8402  042666  004037  051406                  JSR     R0,Q.PUSH         ;PUT PARAMETER BLOCK IN COMMAND
8403  042672  001324                          CINITQ                    ;  INITIATION QUEUE
8404  042674  000207                          RTS     PC                ;RETURN
8405
8406  042676  004737  022642          30$:    JSR     PC,BUFREL         ;RELEASE BUFFER
8407  042702  005765  000146                  TST     P.RECT(R5)        ;DETERMINE IF CURRENTLY UNDERGOING ERROR
8408                                                    ;  RECOVERY
8409  042706  001403                          BEQ     35$               ;YES, PUT IN AVAILIABLE QUEUE
8410  042710  004737  036276                  JSR     PC,SURECY         ;INDICATE SUCCESSFUL RECOVERY
8411  042714  000207                          RTS     PC                ;RETURN
8412
8413  042716  004037  051406          35$:    JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN
8414  042722  001314                          AVAILQ                    ;  AVAILIABLE QUEUE
8415  042724  004737  022342                  JSR     PC,GETBUF         ;GET NEXT BUFFER
8416  042730  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR BEING PROCESSED
8417  042734  000207                          RTS     PC                ;RETURN
8418
8419  042736  042765  100200  000150  40$:    BIC     #BIT15!BIT7,P.DSTT(R5) ;CLEAR FIRST ERROR AND ERROR ENQUEUED
8420  042744  122737  000010  000126          CMPB    #8.,P.RSEC        ;CHECK IF ALL SECTORS HAVE BEEN READ
8421  042752  101422                          BLOS    45$               ;YES, DROP DRIVE FROM TEST SEQUENCE
8422  042754  116565  000123  000135          MOVB    P.RCMD(R5),P.LCMD(R5) ;STORE PREVIOUS COMMAND
8423  042762  016565  000124  000136          MOV     P.RCYL(R5),P.LCYL(R5)
8424  042770  016565  000126  000140          MOV     P.RSEC(R5),P.LSEC(R5)
8425  042776  016537  000130  000142          MOV     P.RWC(R5),P.LWC
8426  043004  004037  051406                  JSR     R0,Q.PUSH         ;ENQUEUE PARAMETER BLOCK IN COMMAND
8427  043010  001324                          CINITQ                    ;  INITIATION QUEUE
8428  043012  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR RECOVERY IN PROGRESS
8429  043016  000207                          RTS     PC                ;RETURN
8430
8431  043020  004737  022642          45$:    JSR     PC,BUFREL         ;GO RELEASE BUFFER
8432  043024  104401  064012                  TYPE    ,ERR204
8433  043030  004737  013004                  JSR     PC,DROP           ;DROP DRIVE FROM TEST SEQUENCE
8434  043034  004737  022342                  JSR     PC,GETBUF         ;GET NEW BUFFER
8435  043040  005037  001566                  CLR     ERRPRO            ;CLEAR ERROR RECOVERY IN PROGRESS
8436  043044  000207                          RTS     PC                ;RETURN
8437
8438  043046  052765  000004  000212  50$:    BIS     #BIT2,P.ERR(R5)   ;SET DATA TYPE ERROR
8439  043054  000137  035310                  JMP     E10$              ;GO PROCESS DATA ERROR
```

```
8440                                        .SBTTL  ERROR RECOVERY SEQUENCE ABNORMAL RETURN
8441
8442  043060  032765  000120  000150  ERCVY2: BIT     #BIT4!BIT6,P.DSTT(R5) ;CHECK IF DIAGNOSING SEQUENCE
8443  043066  001524                          BEQ     35$             ;NO, GO TO ERROR RECOVERY SEQUENCE
8444  043070  004737  030000                  JSR     PC,PRI000       ;PRINT HEADER
8445  043074  104401  064201                  TYPE    ,ERR208         ;TYPE "ERROR WHILE DIAGNOSING ---"
8446  043100  032765  000020  000150          BIT     #BIT4,P.DSTT(R5) ;CHECK IF HEADER TYPE ERROR
8447  043106  001005                          BNE     2$              ;YES, PRINT MESSAGE
8448  043110  104401  062436                  TYPE    ,ERR052+1       ;TYPE "TIME OUT WHILE DRIVE POSITION"
8449  043114  104401  001165                  TYPE    ,$CRLF          ;TYPE <CR><LF>
8450  043120  000402                          BR      3$              ;GO PRINT STATUS
8451
8452  043122  104401  064233          2$:     TYPE    ,ERR209+1       ;PRINT HEADER TYPE ERROR
8453  043126  032777  020000  136004  3$:     BIT     #SW13,@SWR      ;CHECK IF INHIBIT PRINT OUT
8454  043134  001066                          BNE     30$             ;YES, DO NOT PRINT STATUS
8455  043136  032765  000020  000212          BIT     #BIT4,P.ERR(R5) ;CHECK IF DRIVE SEIZED TIME OUT
8456  043144  001402                          BEQ     4$
8457  043146  104401  062753                  TYPE    ,ERR059
8458  043152  032765  000040  000212  4$:     BIT     #BIT5,P.ERR(R5) ;CHECK IF TIME OUT WHILE DRIVE POSITIONING
8459  043160  001402                          BEQ     5$
8460  043162  104401  062436                  TYPE    ,ERR052+1
8461  043166  032765  000010  000014  5$:     BIT     #UEXATT,P.PRST(R5) ;CHECK IF UNEXPECTED ATTENTION
8462  043174  001402                          BEQ     6$
8463  043176  104401  062032                  TYPE    ,ERR030
8464  043202  032765  000020  000014  6$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF DRIVE HARD ERROR
8465  043210  001402                          BEQ     7$
8466  043212  104401  061723                  TYPE    ,ERR027
8467  043216  032765  000040  000014  7$:     BIT     #DRVDSC,P.PRST(R5) ;CHECK IF DSC DID NOT CLEAR
8468  043224  001402                          BEQ     8$
8469  043226  104401  061742                  TYPE    ,ERR028
8470  043232  032765  004000  000014  8$:     BIT     #NODSC,P.PRST(R5) ;CHECK IF ATTN BUT NO DSC OR FAULT
8471  043240  001402                          BEQ     11$
8472  043242  104401  062676                  TYPE    ,ERR058
8473  043246  004737  026074          11$:    JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
8474  043252  004737  025720                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
8475  043256  004737  030244                  JSR     PC,PRI002       ;PRINT RETRY COMMAND
8476  043262  010546                          MOV     R5,-(SP)        ;STORE R5 ON STACK
8477  043264  012705  004516                  MOV     #PARM10,R5      ;LOAD R5 FOR PREVIOUS STATUS
8478  043270  104401  064252                  TYPE    ,ERR210         ;TYPE "FIRST ERROR INFO"
8479  043274  004737  026074                  JSR     PC,PRISTT       ;PRINT CONTROLLER STATUS
8480  043300  004737  025720                  JSR     PC,PRDSTT       ;PRINT DRIVE STATUS
8481  043304  004737  030244                  JSR     PC,PRI002       ;PRINT INITIAL COMMAND
8482  043310  012605                          MOV     (SP)+,R5        ;RESTORE R5
8483  043312  005265  000206          30$:    INC     P.NER(R5)       ;INCREMENT OTHER ERRORS
8484  043316  004737  031250                  JSR     PC,HLT000       ;CHECK IF HALT ON ERROR
8485  043322  004737  022642                  JSR     PC,BUFREL       ;RELEASE BUFFER
8486  043326  004737  013004                  JSR     PC,DROP         ;DROP DRIVE
8487  043332  004737  022342                  JSR     PC,GETBUF       ;GO ALLOCATE BUFFERS
8488  043336  000405                          BR      40$             ;CHECK IF ERROR QUEUE EMPTY
8489
8490  043340  004737  032002          35$:    JSR     PC,ERCVY        ;GO DO ERROR RECOVERY
8491  043344  005737  001566                  TST     ERRPRO          ;CHECK IF ERROR RECOVERY ON LAST DRIVE
8492                                                                  ;  IS FINISHED
8493  043350  001007                          BNE     45$             ;NO, RETURN
8494  043352  004037  051466          40$:    JSR     R0,Q.POP        ;GET NEXT ELEMENT FROM
8495  043356  001330                          ERRPRQ                  ;  ERROR PROCESSING QUEUE
```

```
 8496  043360  012605                    MOV     (SP)+,R5        ;LOAD PARAMETER BLOCK
 8497  043362  010537  001566            MOV     R5,ERRPRO       ;LOAD CURRENT PROCESSING ERROR
 8498  043366  001364                    BNE     35$             ;IF ERROR PROCESSING QUEUE NOT EMPTY GO
 8499                                                            ;  PROCESS NEXT ERROR
 8500  043370  000207            45$:    RTS     PC              ;RETURN
```

```
8501                                        .SBTTL  DETERMINE IF CLOCK IS PRESENT
8502
8503   043372  012737  043422  000004  CHKCLK: MOV     #5$,ERRVEC      ;SET UP ERROR VECTOR FOR NON-EXISTENT MEMORY
8504   043400  012737  000340  000006          MOV     #PR7,ERRVEC+2   ;LOCK ALL INTERRUPTS
8505   043406  005777  135724                  TST     @$LKCSB         ;CHECK FOR P CLOCK
8506   043412  112737  177777  001374          MOVB    #-1,CLKFLG      ;INDICATE KW11-P PRESENT
8507   043420  000413                          BR      20$             ;LOAD COMMON CLOCK PARAMETERS
8508
8509   043422  022626                  5$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
8510   043424  012737  043446  000004          MOV     #15$,ERRVEC     ;SET UP ERROR VECTOR FOR NON-EXISTENT MEMORY
8511   043432  005777  135704                  TST     @$LKS           ;CHECK FOR L CLOCK
8512   043436  112737  000001  001374          MOVB    #1,CLKFLG       ;SET KW11-L PRESENT
8513   043444  000401                          BR      20$             ;REINSTATE TRAP CATCHER
8514
8515   043446  022626                  15$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
8516   043450  012737  000006  000004  20$:    MOV     #ERRVEC+2,ERRVEC ;REINSTATE TRAP CATCHER
8517   043456  005037  000006                  CLR     ERRVEC+2
8518   043462  000207                          RTS     PC              ;RETURN
```

```
 8519                                           .SBTTL  SET UP CLOCK INTERRUPT
 8520
 8521  043464  105737  001374          CLKINT: TSTB    CLKFLG          ;CHECK IF A CLOCK IS ON SYSTEM
 8522  043470  001431                          BEQ     20$             ;NO, RETURN
 8523  043472  100011                          BPL     5$              ;CHECK IF KW11-L
 8524  043474  013701  001340                  MOV     $LPVEC,R1       ;LOAD VECTOR ADDRESS
 8525  043500  012777  177777  135630          MOV     #-1,@$LKCSB     ;LOAD COUNT BUFFER WITH 1'S
 8526  043506  012777  000135  135620          MOV     #135,@$LKCSR    ;SET COUNT UP, 16 MS, CONT
 8527  043514  000405                          BR      10$             ;LOAD COMMON CLOCK PARAMETERS
 8528
 8529  043516  013701  001344          5$:     MOV     $LLVEC,R1       ;LOAD VECTOR ADDRESS
 8530  043522  012777  000100  135612          MOV     #100,@$LKS      ;SET UP L CLOCK FOR INTERRUPT MODE
 8531  043530  012721  043556          10$:    MOV     #CLOCK,(R1)+    ;LOAD ADDRESS OF INTERRUPT ROUTINE
 8532  043534  012711  000300                  MOV     #PR6,(R1)       ;SET UP FOR PRIORITY 6
 8533  043540  113737  001352  001353          MOVB    HZ,CLKFRQ       ;SET UP CLOCK FREQUENCY
 8534  043546  113737  001360  001375          MOVB    PERINV,PERIOD   ;SET UP COUNTER FOR INITIAL STATISTIC SUMMARY
 8535  043554  000207                  20$:    RTS     PC              ;RETURN
```

```
8536                                        .SBTTL  KW11-L AND KW11-P INTERRUPT HANDLER
8537
8538  043556  105337  001353        CLOCK:  DECB    CLKFRQ          ;CHECK IF ONE SECOND PASTED
8539  043562  001041                        BNE     10$             ;NO, RETURN
8540  043564  113737  001352  001353        MOVB    HZ,CLKFRQ       ;REINITIALIZE SYSTEM CLOCK
8541  043572  005237  001372                INC     SECOND          ;INCREMENT SECOND COUNT
8542  043576  022737  000074  001372        CMP     #60.,SECOND     ;CHECK IF MINUTE HAS OCCURRED
8543  043604  001030                        BNE     10$             ;NO, RETURN
8544  043606  005037  001372                CLR     SECOND          ;CLEAR SECOND
8545  043612  105737  001360                TSTB    PERINV          ;CHECK IF INTERVAL STATISTICS
8546  043616  001411                        BEQ     5$              ;NO, UPDATE MINUTE COUNT
8547  043620  105337  001375                DECB    PERIOD          ;DECREMENT PERIOD COUNT
8548  043624  001006                        BNE     5$              ;CHECK IF END OF PERIOD, IF NOT UPDATE MINUTE COUNT
8549  043626  113737  001360  001375        MOVB    PERINV,PERIOD   ;INITIALIZE PERIOD
8550  043634  112737  177777  001633        MOVB    #-1,STATIS      ;INDICATE TIME FOR STATISTIC TYPE OUT
8551  043642  005237  001370          5$:   INC     MINUTE          ;INCREMENT MINUTE COUNTER
8552  043646  022737  000074  001370        CMP     #60.,MINUTE     ;CHECK IF HOUR HAS OCCURED
8553  043654  001004                        BNE     10$             ;NO, RETURN
8554  043656  005037  001370                CLR     MINUTE          ;CLEAR MINUTE COUNT
8555  043662  005237  001366                INC     HOUR            ;INCREMENT HOUR COUNT
8556  043666  000002                  10$:   RTI                     ;RETURN
```

```
8557                                            .SBTTL  PRINT TIME ROUTINE
8558
8559  043670  105737  001374          PRITIM: TSTB    CLKFLG          ;CHECK IF CLOCK ON SYSTEM
8560  043674  001455                          BEQ     50$             ;NO, RETURN
8561  043676  010046                          MOV     R0,-(SP)        ;SAVE R0
8562  043700  013746  177776                  MOV     PS,-(SP)        ;SAVE PSW
8563  043704  012737  000340  177776          MOV     #PR7,PS         ;LOCK OUT CLOCK
8564  043712  013737  001366  001376          MOV     HOUR,TIMHR      ;SAVE TIME FOR PRINT OUT
8565  043720  013737  001370  001400          MOV     MINUTE,TIMMIN
8566  043726  013737  001372  001402          MOV     SECOND,TIMSEC
8567  043734  012637  177776                  MOV     (SP)+,PS        ;RESTORE PSW
8568  043740  012700  064743                  MOV     #TIM001,R0      ;LOAD R0 FOR INDEX
8569  043744  105010                          CLRB    (R0)            ;DETERMINE HUNDREDS OF HOURS
8570  043746  013746  001376                  MOV     TIMHR,-(SP)     ;STORE HOURS FOR CONVERSION
8571  043752  162716  000144          1$:     SUB     #100.,(SP)      ;CONVERT HUNDREDS
8572  043756  100402                          BMI     2$              ;CHECK IF TOO SMALL
8573  043760  105210                          INCB    (R0)
8574  043762  000773                          BR      1$
8575
8576  043764  062716  000144          2$:     ADD     #100.,(SP)
8577  043770  152720  000060                  BISB    #60,(R0)+       ;MAKE DIGIT ASCII
8578  043774  004737  044032                  JSR     PC,CONHUN
8579  044000  013716  001400                  MOV     TIMMIN,(SP)     ;LOAD STACK FOR MINUTES
8580  044004  004737  044032                  JSR     PC,CONHUN
8581  044010  013716  001402                  MOV     TIMSEC,(SP)     ;LOAD STACK FOR SECONDS
8582  044014  004737  044032                  JSR     PC,CONHUN
8583  044020  005726                          TST     (SP)+           ;ADJUST STACK
8584  044022  104401  064736                  TYPE    ,TIM000         ;TYPE TIME:
8585  044026  012600                          MOV     (SP)+,R0        ;RESTORE R0
8586  044030  000207                  50$:    RTS     PC              ;RETURN
8587
8588                                            .SBTTL  CONVERT DECIMAL  LESS THAN 100
8589
8590  044032  105010                  CONHUN: CLRB    (R0)            ;ZERO DIGIT
8591  044034  162766  000012  000002  1$:     SUB     #10.,2(SP)      ;CONVERT TENS
8592  044042  100402                          BMI     5$
8593  044044  105210                          INCB    (R0)
8594  044046  000772                          BR      1$
8595
8596  044050  062766  000012  000002  5$:     ADD     #10.,2(SP)
8597  044056  152720  000060                  BISB    #60,(R0)+       ;MAKE ASCII
8598  044062  105010                          CLRB    (R0)            ;ZERO DIGIT
8599  044064  005766  000002                  TST     2(SP)
8600  044070  001404                          BEQ     7$
8601  044072  105210                  6$:     INCB    (R0)            ;CONVERT ONES
8602  044074  005366  000002                  DEC     2(SP)
8603  044100  001374                          BNE     6$
8604  044102  152720  000060          7$:     BISB    #60,(R0)+       ;MAKE ASCII
8605  044106  105720                          TSTB    (R0)+           ;SKIP OVER COLON
8606  044110  000207                          RTS     PC
```

```
                                .SBTTL  RK611/RK06-RK07 UNIBUS DRIVER FOR QUEUED OPERATIONS (REV. 0.11)

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

                                .SBTTL   *WATCH-DOG TIMER

                                ;;*****************************************************************
                                ;*
                                ;*      THE WATCH-DOG TIMER DOES A PSEUDO-TIMING OF RK06/07 UNIBUS
                                ;*      SUBSYSTEM COMMAND. SINCE ONE CAN NOT GUARANTEE THAT A
                                ;*      REAL-TIME CLOCK (KW11-P OR KW11-L) IS ON THE SYSTEM
                                ;*      THE RK06 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.
                                ;*
                                ;;*****************************************************************

 044112  010546          W.WTCH: MOV     R5,-(SP)        ;SAVE R5 ON THE STACK
 044114  010446                  MOV     R4,-(SP)        ;SAVE R4 ON THE STACK
 044116  010346                  MOV     R3,-(SP)        ;SAVE R3 ON THE STACK
 044120  010246                  MOV     R2,-(SP)        ;SAVE R2 ON STACK
 044122  013746  177776          MOV     PS,-(SP)        ;SAVE PROGRAM STATUS WORD ON STACK
 044126  005337  001466          DEC     W.MTIM          ;DECREMENT MILLISECOND TIMER
 044132  001046                  BNE     20$             ;IF NOT ZERO RETURN
 044134  013737  001470  001466  MOV     W.MILI,W.MTIM   ;REINITIALIZE MILLISECOND TIMER
 044142  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 044150  013702  001446          MOV     RKBAS,R2        ;LOAD BASE OF RK06 REGISTERS
 044154  005004                  CLR     R4              ;CLEAR INDEX OF DRIVE MASKS
 044156  005003                  CLR     R3              ;CLEAR INDEX OF DRIVE TIMES
                                                         ;  AND PARAMETER BLOCK ADDRESSES
 044160  136437  001514  001512 1$:     BITB    INTMSK(R4),W.TIME ;CHECK IF DRIVE IS BEING TIMED
```

```
 8663  044166  001422                        BEQ     6$                ;NO, GO TO NEXT DRIVE
 8664  044170  005363  001544                DEC     W.DRV(R3)         ;DECREMENT DRIVE COUNT
 8665  044174  001017                        BNE     6$                ;IF NOT TIME OUT, GO TO
 8666                                                                  ;  NEXT DRIVE
 8667  044176  016305  001524                MOV     PBLKT(R3),R5      ;LOAD ADDRESS OF PARAMETER
 8668                                                                  ;  BLOCK TABLE FOR INDEXING
 8669  044202  146437  001514  001512        BICB    INTMSK(R4),W.TIME ;RESET TIMING INDICATOR FOR DRIVE
 8670  044210  052765  000100  000014        BIS     #CMDTO,P.PRST(R5) ;SET COMMAND TIME OUT
 8671  044216  020537  001464                CMP     R5,0.WAIT         ;CHECK IF DRIVER IS WAITING FOR
 8672                                                                  ;  COMMAND COMPLETION
 8673  044222  001002                        BNE     5$                ;NO, DO NOT ALTER WAITING FOR
 8674                                                                  ;  COMMAND COMPLETION
 8675  044224  005037  001464                CLR     0.WAIT            ;CLEAR WAIT FOR COMMAND COMPLETION
 8676  044230  004777  135222          5$:   JSR     PC,@A.ABNL        ;BRANCH TO ERROR ROUTINE
 8677  044234  062703  000002          6$:   ADD     #2,R3             ;ADD 2 TO INDEX OF DRIVE TIMES
 8678                                                                  ;  AND PARAMETER BLOCK ADDRESSES
 8679  044240  005204                        INC     R4                ;INCREMENT INDEX OF DRIVE MASK
 8680  044242  022704  000010                CMP     #10,R4            ;CHECK IF ALL DRIVES ARE TESTED
 8681  044246  001344                        BNE     1$                ;NO, CHECK NEXT DRIVE
 8682  044250  012637  177776          20$:  MOV     (SP)+,PS          ;RESTORE PSW
 8683  044254  012602                        MOV     (SP)+,R2          ;RESTORE R2
 8684  044256  012603                        MOV     (SP)+,R3          ;RESTORE R3
 8685  044260  012604                        MOV     (SP)+,R4          ;RESTORE R4
 8686  044262  012605                        MOV     (SP)+,R5          ;RESTORE R5
 8687  044264  000207                        RTS     PC                ;RETURN
```

```
                        .SBTTL   *RK06-RK07 INTERRUPT SERVICE ROUTINE

                        ;;**********************************************************************
                        ;;*
                        ;;*      THIS ROUTINE WILL SERVICE ALL RK06/07 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 WI_L 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
                        ;;*
                        ;;**********************************************************************

044266  010546          I.INTR: MOV     R5,-(SP)        ;STORE R5 ON THE STACK
044270  010446                  MOV     R4,-(SP)        ;STORE R4 ON THE STACK
044272  010346                  MOV     R3,-(SP)        ;STORE R3 ON THE STACK
044274  010246                  MOV     R2,-(SP)        ;STORE R2 ON THE STACK
044276  010146                  MOV     R1,-(SP)        ;STORE R1 ON THE STACK
044300  010046                  MOV     R0,-(SP)        ;STORE R0 ON THE STACK
044302  013702  001446          MOV     RKBAS,R2        ;LOAD R2 TO ADDRESS RK06 REGISTER
044306  016237  000010  001412  MOV     RKCS2(R2),T.CS2 ;STORE CS2
044314  032737  001000  001412  BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
044322  001407                  BEQ     1$              ;NO, CONTINUE PROCESSING
044324  052737  100000  001462  BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT
044332  004777  135122          JSR     PC,@A.CONT      ;REPORT ERROR
```

```
 8744  044336  000137  046674                    JMP     I.RTRN          ;RETURN
 8745
 8746  044342  105737  001506          1$:       TSTB    I.ISRL          ;CHECK IF INTERRUPT OR RELEASE
 8747  044346  001410                             BEQ     6$              ;NO, CHECK IF DRIVE AVAILABLE
 8748  044350  100403                             BMI     5$              ;CHECK IF RELEASE COMMAND
 8749  044352  105037  001506                    CLRB    I.ISRL          ;YES,CLEAR FLAG
 8750  044356  000425                             BR      I.IGO           ;CONTINUE PROCESSING INTERRUPT
 8751
 8752  044360  105037  001506          5$:       CLRB    I.ISRL          ;CLEAR FLAG
 8753  044364  000137  045420                    JMP     I.ATTN          ;GO PROCESS DRIVE ATTENTIONS
 8754
 8755  044370  032737  010400  001412  6$:       BIT     #NED!UFE,T.CS2  ;CHECK FOR NON-EXISTENT DRIVE OR
 8756                                                                    ;  UNIT FIELD ERROR
 8757  044376  001415                             BEQ     7$              ;NO, WAIT FOR DUAL ACCESS INTERRUPT
 8758  044400  013704  001412                    MOV     T.CS2,R4        ;LOAD R4 FOR DRIVE NUMBER
 8759  044404  042704  177770                    BIC     #^C<DRVMSK>,R4  ;KEEP DRIVE BITS
 8760  044410  010403                             MOV     R4,R3           ;STORE DRIVE NUMBER FOR INDEX
 8761  044412  006303                             ASL     R3              ;MULTIPLY BY 2
 8762  044414  016305  001524                    MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS
 8763  044420  016237  000000  001410             MOV     RKCS1(R2),T.CS1 ;LOAD TEMPORARY CS1 FOR STATUS REPORT
 8764  044426  000137  044602                    JMP     I.ERRC          ;REPORT ERROR
 8765
 8766  044432                          7$:
 8767
 8768  044432  013705  001464          I.IOO:    MOV     O.WAIT,R5       ;LOAD PARAMETER BLOCK ADDRESS INTO R5
 8769  044436  001002                             BNE     2$              ;IS COMMAND WAITING PROCESSING
 8770                                                                    ;  YES, DO PROCESSING
 8771  044440  000137  045420                    JMP     I.ATTN          ;NO, PROCESS ATTENTION
 8772
 8773  044444  013704  001412          2$:       MOV     T.CS2,R4        ;STORE RKCS2 FOR DRIVE NUMBER
 8774  044450  042704  177770                    BIC     #^C<DRVMSK>,R4  ;MASK OUT UNNECESSARY BITS
 8775
 8776
 8777  044454  010403                             MOV     R4,R3           ;STORE DRIVE NUMBER FOR INDEX
 8778  044456  006303                             ASL     R3              ;MULTIPLY BY 2 FOR INDEX
 8779
 8780  044460  126504  000000                    CMPB    P.DRVN(R5),R4   ;CHECK IF DRIVE NUMBER IS EXPECTED
 8781  044464  001401                             BEQ     3$              ;YES, CONTINUE
 8782  044466  000000                             HALT                    ;NO, DRIVER ERROR
 8783  044470  122765  000164  000001  3$:       CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
 8784  044476  001002                             BNE     10$             ;NO, EXECUTE NORMAL DATA TRANSFER
 8785  044500  000137  045060                    JMP     I.HDAL          ;GO EXECUTE SPECIAL HEADER SEQUENCE
 8786
 8787  044504  005037  001464          10$:      CLR     O.WAIT          ;CLEAR WAIT FOR COMMAND COMPLETION
 8788  044510  005063  001544                    CLR     W.DRV(R3)       ;CLEAR WATCH-DOG TIME
 8789  044514  146437  001514  001512             BICB    INTMSK(R4),W.TIME ;RESET TIMING ON THIS DRIVE
 8790  044522  016237  000000  001410             MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
 8791  044530  032737  100000  001410             BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 8792  044536  001021                             BNE     I.ERRC          ;YES, PROCESS ERROR
 8793  044540  016237  000016  001424             MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
 8794  044546  136437  001514  001425             BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF DRIVE ATTENTION SET
 8795  044554  001004                             BNE     15$             ;YES, REPORT ERROR
 8796  044556  004777  134672                    JSR     PC,@A.NORM      ;INDICATE NORMAL RETURN
 8797  044562  000137  046674                    JMP     I.RTRN          ;RESTORE REGISTERS
 8798
 8799  044566  052765  000010  000014  15$:      BIS     #UEXATT,P.PRST(R5) ;SET UNEXPECTED ATTENTION
```

```
 8800
 8801  044574  004737  047364          I.ERRA: JSR     PC,I.CSTS       ;STORE CONTROLLER STATUS
 8802  044600  000405                          BR      I.ERR           ;STORE PATTERN AND POSITION INFORMATION
 8803
 8804  044602  013765  001410  000016  I.ERRC: MOV     T.CS1,P.CS1(R5) ;GET ERROR RKCS1
 8805  044610  004737  047406                  JSR     PC,I.CSTI       ;GET REST OF CONTROLLER STATUS
 8806  044614  016265  000032  000062  I.ERR:  MOV     RKECPT(R2),P.EPAT(R5) ;STORE ECC PATTERN
 8807  044622  016265  000030  000060          MOV     RKECPS(R2),P.EPOS(R5) ;STORE ECC POSITION
 8808  044630  004037  046712                  JSR     R0,I.CCLR       ;CLEAR CONTROLLER
 8809  044634  046674                          I.RTRN                  ;ERROR RETURN
 8810  044636  032765  010400  000020          BIT     #NED!UFE,P.CS2(R5) ;CHECK IF IT WAS NON-EXISTENT DRIVE OR
 8811                                                                  ;  UNIT FIELD ERROR
 8812  044644  001056                          BNE     5$              ;YES, REPORT ERROR
 8813  044646  032765  000001  000036          BIT     #DRA,P.DS(R5)   ;SEE IF DRIVE IS AVAIL
 8814  044654  001004                          BNE     4$              ;BR IF YES
 8815  044656  052765  010000  000014          BIS     #DRVSZD,P.PRST(R5) ;SET "DRIVE SIEZED" BIT
 8816  044664  000471                          BR      10$             ;REPORT ERROR
 8817  044666  004037  047470          4$:     JSR     R0,I.STAT       ;GATHER DRIVE STATUS
 8818  044672  046674                          I.RTRN                  ;ERROR RETURN
 8819  044674  112737  000005  001410          MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
 8820  044702  004037  046774                  JSR     R0,I.ISSU       ;ISSUE DRIVE CLEAR
 8821  044706  046674                          I.RTRN                  ;ERROR RETURN
 8822  044710  136437  001514  001425          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
 8823  044716  001407                          BEQ     2$              ;NO, INDICATE DRIVE ERROR
 8824  044720  052737  000020  001462          BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
 8825                                                                  ;  WITH CLEAR
 8826  044726  004777  134526                  JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
 8827  044732  000137  046674                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8828
 8829  044736  032737  040000  001434  2$:     BIT     #S.DSC,T.MR2    ;CHECK IF DRIVE STATUS CHANGE CLEARED
 8830  044744  001403                          BEQ     3$              ;YES, CHECK FAULT
 8831  044746  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DSC DID NOT CLEAR
 8832  044754  032737  001000  001436  3$:     BIT     #S.PAR,T.MR3    ;CHECK IF DRIVE PARITY ERROR
 8833  044762  001407                          BEQ     5$              ;NO, INDICATE ABNORMAL TERMINATION
 8834  044764  052737  002000  001462          BIS     #E.DPAR,E.CONT  ;SET DRIVE PARITY ERROR
 8835  044772  004777  134462                  JSR     PC,@A.CONT      ;INDICATE CONTROLLER ERROR
 8836  044776  000137  046674                  JMP     I.RTRN          ;RETURN
 8837
 8838  045002  032765  000020  000014  5$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF HARD DRIVE ERROR
 8839  045010  001017                          BNE     10$             ;YES, GO REPORT ERROR
 8840  045012  032737  020000  001434          BIT     #S.PIP,T.MR2    ;CHECK IF DRIVE IS CYCLING DOWN
 8841  045020  001413                          BEQ     10$             ;NO, REPORT ERROR
 8842  045022  052765  020000  000014          BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING
 8843  045030  156437  001514  001512          BISB    INTMSK(R4),W.TIME ;SET UP 8 SECONDS FOR
 8844  045036  013763  001474  001544          MOV     W.8SEC,W.DRV(R3) ;  DRIVE TO CYCLE UP
 8845  045044  000137  046674                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8846
 8847  045050  004777  134402          10$:    JSR     PC,@A.ABNL      ;GO REPORT ERROR
 8848  045054  000137  046674                  JMP     I.RTRN          ;GO RESTORE REGISTERS
 8849
 8850                                  .SBTTL   *READ ALL HEADERS INTERRUPT SEQUENCE
 8851
 8852  045060  016237  000000  001410  I.HDAL: MOV     RKCS1(R2),T.CS1 ;STORE CS1 TO CHECK CONTROLLER
 8853                                                                  ;  ERROR
 8854  045066  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR
 8855  045074  001423                          BEQ     5$              ;NO, CHECK FOR ATTENTION
```

```
8856
8857  045076  005037  001464                  CLR     O.WAIT              ;CLEAR WAITING FOR COMMAND COMPLETE
8858  045102  146437  001514  001512          BICB    INTMSK(R4),W.TIME   ;RESET TIMING ON DRIVE
8859  045110  005063  001544                  CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT
8860  045114  013765  001410  000016          MOV     T.CS1,P.CS1(R5)     ;STORE ERROR RKCS1
8861  045122  004737  047406                  JSR     PC,I.CST1           ;STORE CONTROLLER REGISTERS
8862  045126  004037  046712                  JSR     R0,I.CCLR           ;CLEAR CONTROLLER
8863  045132  046674                          I.RTRN                      ;ERROR RETURN
8864  045134  004777  134316                  JSR     PC,@A.ABNL          ;INDICATE ERROR RETURN
8865  045140  000137  046674                  JMP     I.RTRN              ;RESTORE REGISTERS
8866
8867  045144  016237  000016  001424  5$:     MOV     RKASOF(R2),T.ASOF   ;STORE ATTENTION SUMMARY
8868  045152  136437  001514  001425          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF DRIVE ATTENTION IS SET
8869  045160  001411                          BEQ     7$                  ;NO, CHECK IF READ ALL HEADERS
8870  045162  005037  001464                  CLR     O.WAIT              ;CLEAR WAITING FOR COMMAND COMPLETION
8871  045166  146437  001514  001512          BICB    INTMSK(R4),W.TIME   ;RESET TIMING ON DRIVE
8872  045174  005063  001544                  CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT
8873  045200  000137  044574                  JMP     I.ERRA              ;GO REPORT ERROR
8874
8875  045204  013701  001502          7$:     MOV     HDR.AD,R1           ;GET MAIN MEMORY ADDRESS
8876  045210  016221  000024                  MOV     RKDB(R2),(R1)+      ;GET FIRST WORD OF HEADER
8877  045214  016221  000024                  MOV     RKDB(R2),(R1)+      ;GET SECOND WORD OF HEADER
8878  045220  016221  000024                  MOV     RKDB(R2),(R1)+      ;GET THIRD WORD OF HEADER
8879  045224  010137  001502                  MOV     R1,HDR.AD           ;STORE ADDRESS FOR NEXT HEADER
8880  045230  016237  000010  001412          MOV     RKCS2(R2),T.CS2     ;STORE CS2 TO CHECK FOR DATA LATE
8881  045236  032737  100000  001412          BIT     #DLT,T.CS2          ;CHECK FOR DATA LATE
8882  045244  001056                          BNE     35$                 ;YES, REPORT ERROR
8883  045246  005337  001504                  DEC     HDR.CT              ;DECREMENT NUMBER OF HEADER YET TO READ
8884  045252  001027                          BNE     25$                 ;IF NON-ZERO, GO ISSUE NEXT READ HEADER
8885  045254  005037  001464                  CLR     O.WAIT              ;CLEAR DRIVER WAITING FOR COMMAND COMPLETION
8886  045260  005063  001544                  CLR     W.DRV(R3)           ;CLEAR TIME OUT COUNT FOR THIS DRIVE
8887  045264  146437  001514  001512          BICB    INTMSK(R4),W.TIME   ;CLEAR WATCH DOG TIME ON THIS DRIVE
8888  045272  012762  000003  000026          MOV     #3,RKMR1(R2)        ;LOAD MAINTENANCE REGISTER FOR SECTOR COUNT
8889  045300  112737  000001  001410          MOVB    #DR.SEL,T.CS1       ;LOAD SELECT COMMAND
8890  045306  004037  046774                  JSR     R0,I.ISSU           ;GET SECTOR COUNT
8891  045312  046674                          I.RTRN                      ;ERROR RETURN
8892  045314  013765  001436  000056          MOV     T.MR3,P.B11(R5)     ;LOAD SECTOR COUNT
8893  045322  004777  134126                  JSR     PC,@A.NORM          ;INDICATE NORMAL TERMINATION
8894  045326  000137  046674                  JMP     I.RTRN              ;RESTORE REGISTERS
8895
8896  045332  016562  000002  000020  25$:     MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
8897  045340  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
8898  045346  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
8899  045354  042765  165777  000016          BIC     #^C<CDT!CFMT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT AND
8900                                                                      ;  DRIVE TYPE
8901  045362  112765  000125  000016          MOVB    #RDHEAD,P.CS1(R5)   ;STORE COMMAND ISSUED
8902  045370  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE READ HEADER
8903  045376  000137  046674                  JMP     I.RTRN              ;RESTORE REGISTERS
8904
8905  045402  052737  000400  001462  35$:     BIS     #E.DLT,E.CONT       ;SET DATA LATE WHILE UNLOADING HEADER
8906  045410  004777  134044                  JSR     PC,@A.CONT          ;REPORT ERROR
8907  045414  000137  046674                  JMP     I.RTRN              ;RESTORE REGISTERS
8908
8909                                  .SBTTL  *DRIVE ATTENTION SCANNER
8910
8911  045420  016237  000000  001410  I.ATTN: MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS
```

```
8912
8913  045426  032737  100000  001410          BIT     #CERR,T.CS1     ;  REGISTER 1 FOR COMPARISON
8914  045434  001444                          BEQ     5$              ;CHECK IF CONTROLLER ERROR OCCURRED
8915                                                                  ;NO, CHECK IF ATTENTION
8916                                          ;CHECK IF ANY DATA TRANSFER TYPE ERROR EXISTS
8917  045436  032737  164000  001412          BIT     #DLT!WCE!UPE!NEM,T.CS2
8918
8919  045444  001007                          BNE     1$              ;INDICATE ERROR
8920  045446  016237  000014  001426          MOV     RKER(R2),T.ER   ;STORE ERROR REGISTER
8921
8922                                  ;       CHECK FOR DATA TRANSFER ERROR TYPE
8923  045454  032737  125700  001426          BIT     #DCK!OPI!WLE!COE!HVRC!BSE!ECH,T.ER
8924
8925  045462  001407                          BEQ     2$              ;NO DATA TRANSFER ERROR
8926
8927  045464  052737  000010  001462  1$:     BIS     #E.UDAT,E.CONT  ;SET UNEXPECTED DATA TYPE ERROR
8928  045472  004777  133762                  JSR     PC,@A.CONT      ;REPORT ERROR
8929  045476  000137  046674                  JMP     I.RTRN          ;RESTORE REGISTERS
8930
8931  045502  013704  001412          2$:     MOV     T.CS2,R4        ;SAVE CS2 FOR REGISTER NUMBER
8932  045506  042704  177770                  BIC     #^C<DRVMSK>,R4  ;STRIP OFF JUNK
8933  045512  010403                          MOV     R4,R3           ;STORE DRIVE NUMBER IN R3
8934  045514  006303                          ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
8935  045516  146437  001514  001512          BICB    INTMSK(R4),W.TIME ;CLEAR WATCH-DOG TIMER FOR DRIVE
8936  045524  005063  001544                  CLR     W.DRV(R3)       ;RESET WATCH DOG TIME
8937  045530  016305  001524                  MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS IN R5
8938
8939                                  ;       CLEAR DRIVE POSITIONING AND DRIVE POSITIONED FOR DATA TRANSFER
8940                                  ;         IN PROGRAM DEVICE STATUS REGISTER
8941  045534  042765  000006  000014          BIC     #DRVPOS!DRVPDT,P.PRST(R5)
8942
8943  045542  000137  044602                  JMP     I.ERRC          ;GO REPORT ERROR
8944
8945  045546  032737  040000  001410  5$:     BIT     #DI,T.CS1       ;CHECK IF ANY DRIVE ATTENTION
8946  045554  001004                          BNE     6$              ;YES, PROCESS INTERRUPT
8947  045556  004737  051740                  JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
8948  045562  000137  046674                  JMP     I.RTRN          ;RESTORE REGISTERS
8949
8950  045566  016237  000016  001424  6$:     MOV     RKASOF(R2),T.ASOF ;STORE ATTENTION SUMMARY
8951  045574  105737  001425                  TSTB    T.ASOF+1        ;CHECK IF ANY ATTENTIONS SET
8952  045600  001007                          BNE     7$              ;YES GO PROCESS INTERRUPT
8953  045602  052737  000002  001462          BIS     #E.NOAT,E.CONT  ;SET NO ATTENTION IN ATTENTION SUMMARY
8954  045610  004777  133644                  JSR     PC,@A.CONT      ;GO REPORT ERROR
8955  045614  000137  046674                  JMP     I.RTRN          ;GO RESTORE REGISTERS
8956
8957  045620  005004                  7$:     CLR     R4              ;CLEAR DRIVE NUMBER
8958  045622  136437  001514  001425  8$:     BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION OF THIS DRIVE
8959  045630  001002                          BNE     9$              ;YES, PROCESS INTERRUPT
8960  045632  005204                          INC     R4              ;INCREMENT DRIVE NUMBER
8961  045634  000772                          BR      8$              ;CHECK ATTENTION ON NEXT DRIVE
8962
8963  045636  010403                  9$:     MOV     R4,R3           ;STORE DRIVE NUMBER
8964  045640  006303                          ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
8965  045642  016305  001524                  MOV     PBLKT(R3),R5    ;STORE PARAMETER BLOCK ADDRESS
8966  045646  032765  010000  000014          BIT     #DRVSZD,P.PRST(R5) ;CHECK IF DRIVE WAS SEIZED
8967  045654  001402                          BEQ     10$             ;NO, PROCESS NORMAL ATTENTION
```

```
 8968  045656  000137  046506                    JMP     I.DUAL          ;PROCESS THE RELEASE OF DRIVE
 8969
 8970  045662                          10$:
 8971  045662  032765  020000  000014            BIT     #E.UNLD,P.PRST(R5)  ;CHECK IF DRIVE UNLOADING
 8972  045670  001402                            BEQ     11$             ;NO, CONTINUE
 8973  045672  000137  046402                    JMP     I.UNLD          ;SERVICE DRIVE IN POSITION AFTER ERROR
 8974
 8975  045676  042765  000002  000014  11$:     BIC     #DRVPOS,P.PRST(R5)  ;RESET DRIVE POSITIONING
 8976  045704  005062  000026                    CLR     RKMR1(R2)       ;CLEAR MAINTENANCE REGISTER 1
 8977  045710  112737  000001  001410            MOVB    #DR.SEL,T.CS1   ;LOAD COMMAND
 8978  045716  004037  046774                    JSR     R0,I.ISSU       ;SELECT DRIVE WITH ATTENTION HIGH
 8979  045722  046674                            I.RTRN                  ;ERROR RETURN
 8980  045724  013765  001436  000042            MOV     T.MR3,P.B00(R5) ;STORE STATUS BYTE 00 MESS B
 8981  045732  032765  000200  000042            BIT     #S.FLT,P.B00(R5)  ;CHECK IF DRIVE FAULT
 8982  045740  001401                            BEQ     12$             ;NO, CHECK FOR DRIVE STATUS CHANGE
 8983  045742  000473                            BR      I.AERR          ;PROCESS ERROR
 8984
 8985  045744  013765  001434  000040  12$:     MOV     T.MR2,P.A00(R5) ;STORE MAINTENANCE REGISTER 2
 8986  045752  032765  040000  000040            BIT     #S.DSC,P.A00(R5)  ;CHECK FOR DRIVE STATUS CHANGE
 8987  045760  001004                            BNE     13$             ;YES, PROCESS DRIVE STATUS CHANGE
 8988  045762  052765  004000  000014            BIS     #NODSC,P.PRST(R5)  ;SET NO DRIVE STATUS CHANGE
 8989  045770  000460                            BR      I.AERR          ;PROCESS ERROR
 8990
 8991  045772  112737  000005  001410  13$:     MOVB    #DR.CLR,T.CS1   ;LOAD COMMAND
 8992  046000  004037  046774                    JSR     R0,I.ISSU       ;CLEAR DRIVE STATUS CHANGE
 8993  046004  046674                            I.RTRN                  ;ERROR RETUN
 8994  046006  013765  001424  000032            MOV     T.ASOF,P.ASOF(R5)  ;STORE ATTENTION SUMMARY
 8995  046014  136465  001514  000033            BITB    INTMSK(R4),P.ASOF+1(R5)  ;CHECK IF ATTENTION RESET
 8996  046022  001407                            BEQ     15$             ;YES, CONTINUE INTERRUPT PROCESSING
 8997  046024  052737  000020  001462            BIS     #E.CLAT,E.CONT  ;SET ATTENTION DID NOT RESET
 8998                                                                    ;  WITH DRIVE CLEAR
 8999  046032  004777  133422                    JSR     PC,@A.CONT      ;FLAG ERROR
 9000  046036  000137  046674                    JMP     I.RTRN          ;RESTORE REGISTERS
 9001
 9002  046042  013765  001434  000040  15$:     MOV     T.MR2,P.A00(R5)  ;STORE MAINTENANCE REGISTER 2
 9003  046050  032765  040000  000040            BIT     #S.DSC,P.A00(R5)  ;CHECK IF DRIVE STATUS CHANGE
 9004                                                                    ;  RESET
 9005  046056  001404                            BEQ     16$             ;YES, CONTINUE INTERRUPT PROCESSING
 9006  046060  052765  000040  000014            BIS     #DRVDSC,P.PRST(R5)  ;SET DRIVE STATUS CHANGE DID NOT CLEAR
 9007  046066  000421                            BR      I.AERR          ;GO PROCESS ERROR
 9008
 9009  046070  146437  001514  001512  16$:     BICB    INTMSK(R4),W.TIME  ;RESET TIMING ON THIS DRIVE
 9010  046076  005063  001544                    CLR     W.DRV(R3)       ;CLEAR DRIVE TIMING COUNT
 9011  046102  032765  000004  000014            BIT     #DRVPDT,P.PRST(R5)  ;CHECK IF DRIVE IS POSITIONING
 9012                                                                    ;  FOR DATA TRANSFER
 9013  046110  001004                            BNE     17$             ;YES, CALL COMMAND OPTIMIZER
 9014  046112  004777  133336                    JSR     PC,@A.NORM      ;REPORT SUCCESSFUL COMMAND
 9015                                                                    ;  COMPLETION
 9016  046116  000137  046674                    JMP     I.RTRN          ;RESTORE REGISTERS
 9017
 9018  046122  004737  051740          17$:     JSR     PC,C.OPT        ;CALL COMMAND OPTIMIZER
 9019  046126  000137  046674                    JMP     I.RTRN          ;RESTORE REGISTERS
 9020
 9021                                  .SBTTL   *ATTENTION ERROR HANDLER
 9022
 9023  046132  042765  000004  000014  I.AERR: BIC     #DRVPDT,P.PRST(R5) ;RESET POSITIONING IN PROGRESS BECAUSE
```

```
                                                        ;  OF DATA TRANSFER
046140 146437 001514 001512         BICB    INTMSK(R4),W.TIME ;CLEAR TIMING FOR THIS DRIVE
046146 005063 001544                CLR     W.DRV(R3)         ;RESET WATCH-DOG TIME
046152 042765 177741 000016         BIC     #177741,P.CS1(R5) ;KEEP COMMAND ISSUED
046160 042737 000036 001410         BIC     #36,T.CS1         ;KEEP CURRENT CONTROLLER STATUS
046166 053765 001410 000016         BIS     T.CS1,P.CS1(R5)   ;MAKE GOOD MESSAGE
046174 013765 001412 000020         MOV     T.CS2,P.CS2(R5)   ;STORE CONTROLLER REGISTERS
046202 013765 001414 000022         MOV     T.WCR,P.WCR(R5)
046210 013765 001416 000024         MOV     T.BA,P.BAR(R5)
046216 013765 001420 000026         MOV     T.DA,P.DTS(R5)
046224 013765 001422 000030         MOV     T.DC,P.DCYL(R5)
046232 013765 001424 000032         MOV     T.ASOF,P.ASOF(R5)
046240 013765 001426 000034         MOV     T.ER,P.ER(R5)
046246 013765 001430 000036         MOV     T.DS,P.DS(R5)
046254 004037 047470                JSR     R0,I.STAT         ;GATHER DRIVE STATUS
046260 046674                       I.RTRN                    ;ERROR RETURN
046262 112737 000005 001410         MOVB    #DR.CLR,T.CS1     ;LOAD COMMAND
046270 004037 046774                JSR     R0,I.ISSU         ;CLEAR DRIVE ERRORS
046274 046674                       I.RTRN                    ;ERROR RETURN
046276 136437 001514 001425         BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
046304 001407                       BEQ     2$                ;YES, FLAG DRIVE ERROR
046306 052737 000020 001462         BIS     #E.CLAT,E.CONT    ;SET ATTENTION DID NOT RESET
046314 004777 133140                JSR     PC,@A.CONT        ;REPORT ERROR
046320 000137 046674                JMP     I.RTRN            ;RESTORE REGISTERS

046324 032765 000020 000014 2$:     BIT     #DRVHRD,P.PRST(R5) ;CHECK IF AHARD DRIVE ERROR
046332 001017                       BNE     10$               ;YES, REPORT ERROR
046334 032737 020000 001434         BIT     #S.PIP,T.MR2      ;CHECK IF DRIVE IS UNLOADING
046342 001413                       BEQ     10$               ;NO, REPORT ERROR
046344 052765 020000 000014         BIS     #E.UNLD,P.PRST(R5) ;SET DRIVE UNLOADING DUE TO ERROR
046352 156437 001514 001512         BISB    INTMSK(R4),W.TIME ;SET TIMING ON THIS DRIVE
046360 013763 001474 001544         MOV     W.8SEC,W.DRV(R3) ;LOAD 8 SECONDS FOR CYCLE UP
046366 000137 046674                JMP     I.RTRN            ;RESTORE REGISTERS

046372 004777 133060        10$:    JSR     PC,@A.ABNL        ;REPORT ERROR
046376 000137 046674                JMP     I.RTRN            ;RESTORE REGISTERS

                            .SBTTL  *ERROR CAUSING DRIVE TO UNLOAD

046402 052765 020000 000014 I.UNLD: BIS     #E.UNLD,P.PRST(R5) ;CLEAR DRIVE UNLOADING BECAUSE OF ERROR
046410 112737 000005 001410         MOVB    #DR.CLR,T.CS1     ;LOAD IN DRIVE CLEAR
046416 004037 046774                JSR     R0,I.ISSU         ;GO ISSUE DRIVE CLEAR
046422 046674                       I.RTRN                    ;ERROR RETURN
046424 136437 001514 001425         BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION CLEARED
046432 001406                       BEQ     15$               ;YES, CONTINUE
046434 012737 000020 001462         MOV     #E.CLAT,E.CONT    ;SET ATTENTION DID NOT RESET
046442 004777 133012                JSR     PC,@A.CONT        ;REPORT ERROR
046446 000512                       BR      I.RTRN            ;RESTORE REGISTERS

046450 032737 040000 001434 15$:    BIT     #S.DSC,T.MR2      ;CHECK IF DRIVE STAUS CHANGE RESET
046456 001403                       BEQ     20$               ;YES, CONTINUE
046460 052765 000040 000014         BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STAUS CHANGE DID NOT CLEAR
046466 146437 001514 001512 20$:    BICB    INTMSK(R4),W.TIME ;RESET TIMING ON THIS DRIVE
046474 005063 001544                CLR     W.DRV(R3)         ;CLEAR TIME COUNT
046500 004777 132752                JSR     PC,@A.ABNL        ;REPORT ERROR
046504 000473                       BR      I.RTRN            ;RESTORE REGISTER
```

```
9080
9081                                        .SBTTL   *DUAL ACCESS INTERRUPT HANDLER
9082
9083  046506  112737  000001  001410  I.DUAL: MOVB    #DR.SEL,T.CS1    ;LOAD COMMAND
9084  046514  004037  046774                  JSR     R0,I.ISSU        ;SELECT DRIVE
9085  046520  046674                          I.RTRN                   ;ERROR RETURN
9086  046522  032737  000200  001436          BIT     #S.FLT,T.MR3     ;CHECK IF DRIVE FAULT
9087  046530  001402                          BEQ     10$              ;NO, PROCESS INTERRUPT
9088  046532  000137  046132                  JMP     I.AERR           ;PROCESS ATTENTION ERROR
9089
9090  046536  032737  040000  001434  10$:    BIT     #S.DSC,T.MR2     ;CHECK IF DRIVE STATUS CHANGE SET
9091  046544  001005                          BNE     11$              ;YES, PROCESS INTERRUPT
9092  046546  052765  004000  000014          BIS     #NODSC,P.PRST(R5) ;SET NO DRIVE STATUS CHANGE
9093  046554  000137  046132                  JMP     I.AERR           ;PROCESS ATTENTION ERROR
9094
9095  046560  042765  010000  000014  11$:    BIC     #DRVSZD,P.PRST(R5) ;RESET DRIVE SEIZED
9096  046566  112737  000005  001410          MOVB    #DR.CLR,T.CS1    ;LOAD COMMAND
9097  046574  004037  046774                  JSR     R0,I.ISSU        ;CLEAR DRIVE DSC
9098  046600  046674                          I.RTRN                   ;ERROR RETURN
9099  046602  136437  001514  001425          BITB    INTMSK(R4),T.ASOF+1 ;CHECK IF ATTENTION RESET
9100  046610  001406                          BEQ     15$              ;YES, CONTINUE
9101  046612  052737  000020  001462          BIS     #E.CLAT,E.CONT   ;SET ATTENTION DID NOT RESET
9102  046620  004777  132634                  JSR     PC,@A.CONT       ;INDICATE CONTROLLER ERROR
9103  046624  000423                          BR      I.RTRN           ;RESTORE REGISTERS
9104
9105  046626  032737  040000  001434  15$:    BIT     #S.DSC,T.MR2     ;CHECK IF DRIVE STATUS CHANGE IS STILL SET
9106  046634  001405                          BEQ     20$              ;NO, GO ENQUEUE COMMAND
9107  046636  052765  000040  000014          BIS     #DRVDSC,P.PRST(R5) ;SET DRIVE STATUS CHANGE DID NOT CLEAR
9108  046644  000137  046132                  JMP     I.AERR           ;REPORT ATTENTION ERROR
9109
9110  046650  146437  001514  001512  20$:    BICB    INTMSK(R4),W.TIME ;STOP TIMING ON THIS DRIVE
9111  046656  005063  001544                  CLR     W.DRV(R3)        ;CLEAR WATCH DOG TIME OF THIS DRIVE
9112  046662  004037  051406                  JSR     R0,Q.PUSH        ;PUT PARAMETER BLOCK ADDRESS ON
9113  046666  001324                          CINITQ                   ;  COMMAND INITATION QUEUE
9114  046670  004737  051740                  JSR     PC,C.OPT         ;CALL COMMAND OPTIMIZER
9115
9116  046674  012600          I.RTRN: MOV     (SP)+,R0         ;RESTORE R0
9117  046676  012601                  MOV     (SP)+,R1         ;RESTORE R1
9118  046700  012602                  MOV     (SP)+,R2         ;RESTORE R2
9119  046702  012603                  MOV     (SP)+,R3         ;RESTORE R3
9120  046704  012604                  MOV     (SP)+,R4         ;RESTORE R4
9121  046706  012605                  MOV     (SP)+,R5         ;RESTORE R5
9122  046710  000002                  RTI                      ;RETURN
9123
```

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

046712  012762  100000  000000  I.CCLR: MOV     #CCLR,RKCS1(R2) ;CLEAR CONTROLLER
046720  016237  000000  001410          MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
046726  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER CLEAR DID
                                                                ;  CLEAR ERROR
046734  001407                        BEQ     5$              ;YES, RETURN TO DRIVER PROCESSING
046736  052737  000001  001462        BIS     #E.CCLR,E.CONT  ;SET CLEAR CONTROLLER DID NOT CLEAR ERROR
046744  004777  132510                JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
046750  011000                        MOV     (R0),R0         ;SET UP ERROR RETURN
046752  000200                        RTS     R0              ;RETURN

046754  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
046762  112737  177777  001506          MOVB    #-1,I.ISRL      ;SET INTERRUPT ENABLE ISSUED
046770  005720                        TST     (R0)+           ;ADJUST FOR NORMAL RETURN
046772  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
                                ;*
                                ;;*****************************************************************

046774  013746  001410          I.ISSU: MOV     T.CS1,-(SP)     ;STORE COMMAND ISSUED
047000  005037  001412                  CLR     T.CS2           ;CLEAR TEMPORARY CS2
047004  116537  000000  001412          MOVB    P.DRVN(R5),T.CS2 ;LOAD IN DRIVE NUMBER
047012  013762  001412  000010          MOV     T.CS2,RKCS2(R2) ;LOAD DRIVE NUMBER FOR COMMAND
047020  116537  000007  001411          MOVB    P.CS1H(R5),T.CS1+1 ;STORE BITS 8-15 OF CS1
047026  142737  177753  001411          BICB    #^C<B.CDT!B.CFMT>,T.CS1+1 ;CLEAR ALL BITS EXCEPT
                                                                ;  FORMAT AND DRIVE TYPE
047034  013762  001410  000000          MOV     T.CS1,RKCS1(R2) ;ISSUE COMMAND
047042  105762  000000          1$:     TSTB    RKCS1(R2)       ;WAIT FOR READY
047046  100375                          BPL     1$
047050  004737  047236                  JSR     PC,I.STOR       ;GO STORE REGISTERS
047054  032737  100000  001410          BIT     #CERR,T.CS1     ;CHECK IF CONTROLLER ERROR OCCUERED
047062  001447                          BEQ     5$              ;NO, RETURN
047064  032737  001000  001412          BIT     #MDS,T.CS2      ;CHECK IF MULTIPLE DRIVE SELECT
047072  001406                          BEQ     2$              ;NO, CHECK FOR OTHER CONTROLLER ERRORS
047074  052737  100000  001462          BIS     #E.MDS,E.CONT   ;SET MULTIPLE DRIVE SELECT FLAG
047102  004777  132352                  JSR     PC,@A.CONT      ;REPORT CONTROLLER ERROR
047106  000450                          BR      10$             ;RETURN

                                ;CHECK IF ANY CONTROLLER ERROR IS SET
047110  022737  000001  001500  2$:     CMP     #1,OPTFLG       ;SEE IF CURRENTLY IN CMND OPTIMIZER
047116  001004                          BNE     4$              ;BR IF NOT
047120  032737  010000  001412          BIT     #NED,T.CS2      ;SEE IF NED ERROR SET
047126  001022                          BNE     3$              ;BR IF YES, TO ISSUE CLEAR AND EXIT
047130  032737  024000  001410  4$:     BIT     #CTO!SPAR,T.CS1
047136  001027                          BNE     7$
047140  032737  176400  001412          BIT     #UFE!PGE!NEM!NED!UPE!WCE!DLT,T.CS2
047146  001023                          BNE     7$
```

```
 9215  047150  032737  131761  001426          BIT     #ILC!DTYE!FMTE!ECH!BSE!HVRC!COE!DTE!OPI!DCK,T.ER
 9216  047156  001017                          BNE     7$
 9217
 9218  047160  122716  000005                  CMPB    #DR.CLR,(SP)    ;CHECK IF CLEAR DRIVE
 9219  047164  001003                          BNE     3$              ;NO, DO NOT SET DRIVE HARD ERROR
 9220  047166  052765  000020  000014          BIS     #DRVHRD,P.PRST(R5) ;SET HARD DRIVE ERROR
 9221  047174  004037  046712          3$:     JSR     R0,I.CCLR       ;GO ISSUE A CONTROLLER CLEAR
 9222  047200  047230                          .10$                    ;ERROR RETURN
 9223  047202  012762  000100  000000  5$:     MOV     #IE,RKCS1(R2)   ;SET INTERRUPT ENABLE
 9224  047210  005726                          TST     (SP)+           ;ADJUST STACK
 9225  047212  005720                          TST     (R0)+           ;ADJUST R0 FOR NORMAL RETURN
 9226  047214  000200                          RTS     R0              ;RETURN
 9227
 9228  047216  052737  001000  001462  7$:     BIS     #E.CERR,E.CONT  ;SET CONTROLLER ERROR DURING
 9229                                                                  ;  DRIVER SERVICING
 9230  047224  004777  132230                  JSR     PC,@A.CONT      ;REPORT ERROR
 9231  047230  005726                  10$:    TST     (SP)+           ;ADJUST STACK
 9232  047232  011000                          MOV     (R0),R0         ;ADJUST R0 FOR ERROR RETURN
 9233  047234  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL   *STORE RK611 UNIBUS REGISTERS

                                ;;*****************************************************************
                                ;*
                                ;*      THIS SUBROUTINE IS CALLED BY THE RK06/07 DRIVER TO STORE ALL
                                ;*      RK611 REGISTER IN TEMPORARY LOCATIONS.
                                ;*
                                ;*CALL  JSR     PC,I.STOR
                                ;*      RETURN
                                ;*
                                ;*      REGISTER        USE
                                ;*      --------        ---
                                ;*
                                ;*      R2              ADDRESS OF RK611 REGISTERS
                                ;*
                                ;;*****************************************************************

 047236  016237  000000  001410  I.STOR: MOV     RKCS1(R2),T.CS1 ;STORE ALL CONTROLLER REGISTERS
 047244  016237  000010  001412          MOV     RKCS2(R2),T.CS2 ;  EXCEPT DATA BUFFER
 047252  016237  000002  001414          MOV     RKWC(R2),T.WCR
 047260  016237  000004  001416          MOV     RKBA(R2),T.BA
 047266  016237  000006  001420          MOV     RKDA(R2),T.DA
 047274  016237  000012  001430          MOV     RKDS(R2),T.DS
 047302  016237  000014  001426          MOV     RKER(R2),T.ER
 047310  016237  000016  001424          MOV     RKASOF(R2),T.ASOF
 047316  016237  000020  001422          MOV     RKDCYL(R2),T.DC
 047324  016237  000026  001432          MOV     RKMR1(R2),T.MR1
 047332  016237  000034  001434          MOV     RKMR2(R2),T.MR2
 047340  016237  000036  001436          MOV     RKMR3(R2),T.MR3
 047346  016237  000030  001440          MOV     RKECPS(R2),T.POS
 047354  016237  000032  001442          MOV     RKECPT(R2),T.PAT
 047362  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                      RK06/07 BASE ADDRESS
                                ;*              R5                      ADDRESS OF PARAMETER BLOCK
                                ;*
                                ;*****************************************************************
047364  042765  177741  000016  I.CSTS: BIC     #177741,P.CS1(R5)       ;CLEAR ALL BITS EXCEPT FUNCTION
                                                                        ;OF LAST COMMAND ISSUED
047372  042737  000036  001410          BIC     #36,T.CS1               ;CLEAR FUNCTION OF CS1 STATUS
047400  053765  001410  000016          BIS     T.CS1,P.CS1(R5)         ;GENERATE CS1 STATUS INFORMATION
047406  016265  000010  000020  I.CST1: MOV     RKCS2(R2),P.CS2(R5)     ;STORE COMMAND AND STATUS REGISTER 2
047414  016265  000002  000022          MOV     RKWC(R2),P.WCR(R5)      ;STORE WORD COUNT REGISTER
047422  016265  000004  000024          MOV     RKBA(R2),P.BAR(R5)      ;STORE BUS ADDRESS REGISTER
047430  016265  000006  000026          MOV     RKDA(R2),P.DTS(R5)      ;STORE DESIRED TRACK AND SECTOR
047436  016265  000012  000036          MOV     RKDS(R2),P.DS(R5)       ;STORE DRIVE STATUS REGISTER
047444  016265  000014  000034          MOV     RKER(R2),P.ER(R5)       ;STORE ERROR REGISTER
047452  016265  000016  000032          MOV     RKASOF(R2),P.ASOF(R5)   ;STORE ATTENTION SUMMARY AND
                                                                        ;  OFFSET
047460  016265  000020  000030          MOV     RKDCYL(R2),P.DCYL(R5)   ;STORE CYLINDER ADDRESS
047466  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
                                ;*
                                ;;*****************************************************************

047470  012762  000001  000026  I.STAT: MOV     #1,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                        ;  FOR STATUS BYTE 01
047476  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047504  004037  046774                  JSR     R0,I.ISSU               ;GET STATUS BYTES 01
047510  047700                          3$                              ;ERROR RETURN
047512  013765  001434  000044          MOV     T.MR2,P.A01(R5)         ;STORE STATUS BYTE 01 MESS A
047520  013765  001436  000046          MOV     T.MR3,P.B01(R5)         ;STORE STATUS BYTE 01 MESS B
047526  012762  000002  000026          MOV     #2,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER 1
                                        ;  FOR STATUS BYTE 10
047534  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047542  004037  046774                  JSR     R0,I.ISSU               ;GET STATUS BYTES 10
047546  047700                          3$                              ;ERROR RETURN
047550  013765  001434  000050          MOV     T.MR2,P.A10(R5)         ;STORE STATUS BYTE 10 MESS A
047556  013765  001436  000052          MOV     T.MR3,P.B10(R5)         ;STORE STATUS BYTE 10 MESS B
047564  012762  000003  000026          MOV     #3,RKMR1(R2)            ;LOAD MAINTENANCE REGISTER
                                        ;  FOR STATUS BYTE 11
047572  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047600  004037  046774                  JSR     R0,I.ISSU               ;GET STATUS BYTES 11
047604  047700                          3$                              ;ERROR RETURN
047606  013765  001434  000054          MOV     T.MR2,P.A11(R5)         ;STORE STATUS BYTE 11 MESS A
047614  013765  001436  000056          MOV     T.MR3,P.B11(R5)         ;STORE STATUS BYTE 11 MESS B
047622  005062  000026                  CLR     RKMR1(R2)               ;LOAD MAINTENANCE REGISTER 1
                                        ;  FOR STATUS BYTE 00
047626  112737  000001  001410          MOVB    #DR.SEL,T.CS1           ;LOAD COMMAND
047634  004037  046774                  JSR     R0,I.ISSU               ;GET STATUS BYTES 00
047640  047700                          3$                              ;ERROR RETURN
047642  013765  001434  000040          MOV     T.MR2,P.A00(R5)         ;STORE STATUS BYTE 00 MESS A
047650  013765  001436  000042          MOV     T.MR3,P.B00(R5)         ;STORE STATUS BYTE 00 MESS B
047656  032737  001000  001436          BIT     #S.PAR,T.MR3            ;CHECK IF BAD PARITY DETECTED BY DRIVE
047664  001407                          BEQ     5$                      ;NO, RETURN NORMALLY
```

```
 9367  047666  052737  002000  001462          BIS     #E.DPAR,E.CONT          ;INDICATE BAD PARITY DETECTED BY DRIVE
 9368  047674  004777  131560                  JSR     PC,@A.CONT              ;REPORT ERROR
 9369  047700  011000                  3$:     MOV     (R0),R0                 ;LOAD R0 FOR ERROR RETURN
 9370  047702  000200                          RTS     R0                      ;RETURN
 9371
 9372  047704  052765  001000  000014  5$:     BIS     #PBSVAL,P.PRST(R5)      ;SET PARAMETER BLOCK STATUS VALID
 9373  047712  005720                          TST     (R0)+                   ;ADJUST R0 FOR NORMAL RETURN
 9374  047714  000200                          RTS     R0                      ;RETURN
 9375
```

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

 047716  010546          C.INIT: MOV     R5,-(SP)        ;STORE R5 ON STACK
 047720  010446                  MOV     R4,-(SP)        ;STORE R4 ON STACK
 047722  010346                  MOV     R3,-(SP)        ;STORE R3 ON STACK
 047724  010246                  MOV     R2,-(SP)        ;STORE R2 ON STACK
 047726  010146                  MOV     R1,-(SP)        ;STORE R1 ON STACK
 047730  010046                  MOV     R0,-(SP)        ;STORE R0 ON STACK
 047732  013746  177776          MOV     PS,-(SP)        ;STORE PSW ON STACK
 047736  013737  001452  177776  MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 047744  017605  000016          MOV     @16(SP),R5      ;STORE PARAMETER BLOCK ADDRESS
 047750  062766  000002  000016  ADD     #2,16(SP)       ;ADJUST RETURN
 047756  016504  000000          MOV     P.DRVN(R5),R4   ;STORE DRIVE NUMBER
 047762  042704  177770          BIC     #^C<DRVMSK>,R4  ;MASK OUT JUNK
 047766  156437  001514  001512  BISB    INTMSK(R4),W.TIME ;SET WATCH DOG TIME
 047774  010403                  MOV     R4,R3           ;STORE DRIVE NUMBER
 047776  006303                  ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
 050000  013763  001472  001544  MOV     W.SEC,W.DRV(R3) ;LOAD WATCH-DOG TIME

 050006  013702  001446          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
```

```
 9432  050012  042765  075176  000014          BIC     #↑C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
 9433
 9434  050020  010500                          MOV     R5,R0           ;STORE PARAMETER BLOCK ADDRESS
 9435  050022  062700  000016                  ADD     #P.CS1,R0       ;CALCULATE FIRST LOCATION TO BE CLEARED
 9436  050026  010501                          MOV     R5,R1           ;STORE PARAMETER BLOCK ADDRESS
 9437  050030  062701  000062                  ADD     #P.EPAT,R1      ;CALCULATE LAST LOCATION TO BE CLEARED
 9438
 9439  050034  005020              1$:         CLR     (R0)+           ;CLEAR RETURN PARAMETER
 9440  050036  020001                          CMP     R0,R1           ;CHECK IF FINISHED
 9441  050040  101775                          BLOS    1$              ;NO, CLEAR NEXT RETURN PARAMETER
 9442  050042  105037  001506                  CLRB    I.ISRL          ;CLEAR RELEASE OR INTERRUPT ISSUED
 9443  050046  010465  000020                  MOV     R4,P.CS2(R5)    ;STORE DRIVE NUMBER
 9444  050052  005062  000026                  CLR     RKMR1(R2)       ;CLEAR RK06 MAINTENANCE REGISTER 1
 9445  050056  132765  000040  000001          BITB    #BIT5,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
 9446  050064  001402                          BEQ     3$              ;NO, PROCESS
 9447  050066  000137  050446                  JMP     C.SPEC          ;JUMP TO SPECIAL COMMAND PROCESSOR
 9448
 9449  050072  122765  000107  000001  3$:     CMPB    #UNLOAD,P.CMND(R5) ;CHECK IF POSITIONING COMMAND
 9450                                                                  ;  START SPINDLE
 9451                                                                  ;  RECALIBRATE
 9452                                                                  ;  OFFSET
 9453                                                                  ;  SEEK
 9454                                                                  ;  UNLOAD
 9455
 9456  050100  101125                          BHI     25$             ;NO, DRIVE COMMAND
 9457                                                                  ;  SELECT DRIVE
 9458                                                                  ;  PACK ACKNOWLEDGE
 9459                                                                  ;  CLEAR
 9460
 9461  050102  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF DATA TRANSFER
 9462  050110  103471                          BLO     20$             ;YES, DATA TRANSFER COMMAND
 9463                                                                  ;  READ DATA
 9464                                                                  ;  WRITE DATA
 9465                                                                  ;  READ HEADER
 9466                                                                  ;  WRITE HEADER
 9467                                                                  ;  WRITE CHECK
 9468  050112  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
 9469  050120  052765  000002  000014          BIS     #DRVPOS,P.PRST(R5) ;SET DRIVE POSITIONING
 9470  050126  005037  001464                  CLR     0.WAIT          ;CLEAR WAIT FOR COMMAND
 9471  050132  122765  000117  000001          CMPB    #SEEK,P.CMND(R5) ;CHECK IF SEEK
 9472  050140  001007                          BNE     5$              ;NO, CHECK FOR OFFSET
 9473  050142  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
 9474  050150  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK
 9475  050156  000431                          BR      8$              ;GO ISSUE COMMAND
 9476
 9477  050160  122765  000115  000001  5$:     CMPB    #OFFSET,P.CMND(R5) ;CHECK IF OFFSET
 9478  050166  001007                          BNE     6$              ;NO, CHECK FOR UNLOAD
 9479  050170  116565  000006  000032          MOVB    P.OFST(R5),P.ASOF(R5) ;STORE OFFSET
 9480  050176  016562  000032  000016          MOV     P.ASOF(R5),RKASOF(R2) ;LOAD OFFSET REGISTER
 9481  050204  000416                          BR      8$              ;GO ISSUE COMMAND
 9482
 9483  050206  122765  000111  000001  6$:     CMPB    #SRTSPL,P.CMND(R5) ;CHECK IF START SPINDLE
 9484  050214  001003                          BNE     7$              ;NO, CHECK IF RECAL
 9485  050216  013763  001476  001544          MOV     W.MIN,W.DRV(R3) ;LOAD WATCH DOG TIME FOR 1 MINUTE
 9486  050224  122765  000113  000001  7$:     CMPB    #RECAL,P.CMND(R5) ;CHECK IF RECAL
 9487  050232  001003                          BNE     8$              ;NO, CONTINUE
```

```
 9488  050234  013763  001474  001544          MOV     W.8SEC,W.DRV(R3) ;LOAD RECAL TIME TO 8 SECONDS
 9489  050242  116565  000007  000017  8$:     MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9490  050250  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 9491                                                  ;  AND DRIVE TYPE
 9492  050256  116565  000001  000016          MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 9493  050264  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 9494  050272  000460                          BR      30$             ;GO, ISSUE COMMAND
 9495
 9496  050274  016562  000010  000004  20$:    MOV     P.BALO(R5),RKBA(R2) ;LOAD BUS ADDRESS REGISTER
 9497  050302  016562  000012  000002          MOV     P.WC(R5),RKWC(R2) ;LOAD WORD COUNT REGISTER
 9498  050310  016562  000002  000020          MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS REGISTER
 9499  050316  016562  000004  000006          MOV     P.SECT(R5),RKDA(R2) ;LOAD SECTOR AND TRACK NUMBER
 9500  050324  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5)  ;CHECK IF WRITE CHECK COMMAND
 9501  050332  001010                          BNE     25$             ;NO, GO ISSUE THE COMMAND
 9502  050334  032765  000200  000014          BIT     #W.WCK,P.PRST(R5)  ;CHECK IF WRITE COMMAND SHOULD BE ISSUED
 9503  050342  001404                          BEQ     25$             ;NO, GO ISSUE THE COMMAND
 9504  050344  012765  000123  000016          MOV     #WRDATA,P.CS1(R5)  ;ISSUE WRITE COMMAND
 9505  050352  000406                          BR      26$             ;GO ISSUE COMMAND
 9506
 9507  050354  116565  000001  000016  25$:    MOVB    P.CMND(R5),P.CS1(R5) ;MOVE COMMAND INTO CS1
 9508  050362  042765  000200  000014          BIC     #W.WCK,P.PRST(R5) ;RESET WRITE FOR WRITE CHECK
 9509  050370  116565  000007  000017  26$:    MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9510  050376  142765  177750  000017          BICB    #^C<B.CFMT!B.CDT!B.BA16!B.BA17>,P.CS1+1(R5) ;CLEAR ALL BITS EXCEPT
 9511                                                  ;  FORMAT, DRIVE TYPE, AND BUS ADDRESS
 9512                                                  ;  BITS 16-17
 9513  050404  010537  001464                  MOV     R5,O.WAIT       ;LOAD WAITING FOR COMMAND
 9514  050410  032765  100000  000014          BIT     #DTBAII,P.PRST(R5) ;CHECK IF INHIBIT BUS ADDRESS INCREMENT
 9515  050416  001403                          BEQ     27$             ;NO, LOAD CS2
 9516  050420  052765  000020  000020          BIS     #BAI,P.CS2(R5)  ;SET INHIBIT BUS ADDRESS INCREMENT
 9517  050426  016562  000020  000010  27$:    MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2
 9518  050434  016562  000016  000000  30$:    MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 9519  050442  000137  051364                  JMP     C.RTRN          ;RESTORE REGISTERS
 9520
 9521                                  .SBTTL   *SPECIAL COMMAND PROCESSING
 9522
 9523  050446  122765  000141  000001  C.SPEC: CMPB    #RDSTAT,P.CMND(R5) ;CHECK IF READ DRIVE STATUS
 9524  050454  001132                          BNE     10$             ;NO, PROCESS OTHER COMMANDS
 9525  050456  016562  000020  000010          MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR COMMAND
 9526  050464  116565  000007  000017          MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9527  050472  042765  165777  000016          BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
 9528                                                  ;  AND DRIVE TYPE
 9529  050500  112765  000001  000016          MOVB    #DR.SEL,P.CS1(R5) ;STORE COMMAND
 9530  050506  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
 9531  050514  004737  044112          2$:     JSR     PC,W.WTCH       ;CALL WATCH-DOG TIMER
 9532  050520  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REG. 1
 9533  050526  032765  000200  000016          BIT     #RDY,P.CS1(R5)  ;WAIT FOR READY
 9534  050534  001767                          BEQ     2$
 9535  050536  004737  047406                  JSR     PC,I.CST1       ;STORE CONTROLLER REGISTERS
 9536  050542  016265  000034  000040          MOV     RKMR2(R2),P.AOO(R5) ;STORE STATUS BYTE OO MESSAGE A
 9537  050550  016265  000036  000042          MOV     RKMR3(R2),P.BOO(R5) ;STORE STATUS BYTE OO MESSAGE B
 9538  050556  032765  100000  000016          BIT     #CERR,P.CS1(R5) ;CHECK IF CONTROLLER ERROR
 9539  050564  001443                          BEQ     6$              ;NO, GATHER DRIVE STATUS
 9540  050566  146437  001514  001512          BICB    INTMSK(R4),W.TIME  ;RESET WATCH DOG TIMING ON THIS DRIVE
 9541  050574  005063  001544                  CLR     W.DRV(R3)       ;CLEAR WATCH DOG COUNT
 9542  050600  032765  001000  000020          BIT     #MDS,P.CS2(R5)  ;CHECK IF MULTIPLE DRIVE SELECT
 9543  050606  001046                          BNE     9$              ;YES, INDICATE CONTROLLER ERROR
```

```
9544  050610 004037 046712                    JSR     R0,I.CCLR          ;CLEAR ERROR
9545  050614 051364                           C.RTRN                     ;ERROR RETURN
9546  050616 032765 010400 000020             BIT     #NED!UFE,P.CS2(R5) ;CHECK IF NON-EXISTENT DRIVE OR UNIT FIELD ERROR
9547  050624 001017                           BNE     5$                 ;REPORT ERROR
9548  050626 032765 000001 000036             BIT     #DRA,P.DS(R5)      ;CHECK IF DRIVE AVAILIABLE
9549  050634 001013                           BNE     5$                 ;YES, REPORT ERROR
9550  050636 156437 001514 001512             BISB    INTMSK(R4),W.TIME  ;SET TIMING FOR THIS DRIVE
9551  050644 013763 001476 001544             MOV     W.MIN,W.DRV(R3)    ;WAIT 1 MINUTE FOR RELEASE
9552  050652 052765 010000 000014             BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SEIZED
9553  050660 000137 051364                    JMP     C.RTRN             ;RESTORE REGISTERS
9554
9555  050664 004777 130566          5$:       JSR     PC,@A.ABNL         ;REPORT ERROR
9556  050670 000137 051364                    JMP     C.RTRN             ;RESTORE REGISTERS
9557
9558  050674 004037 047470          6$:       JSR     R0,I.STAT          ;GATHER DRIVE STATUS
9559  050700 051364                           C.RTRN                     ;ERROR RETURN
9560  050702 146437 001514 001512             BICB    INTMSK(R4),W.TIME  ;STOP WATCH-DOG TIMING ON DRIVE
9561  050710 005063 001544                    CLR     W.DRV(R3)          ;RESET WATCH-DOG TIME
9562  050714 004777 130534                    JSR     PC,@A.NORM         ;REPORT COMMAND COMPLETE
9563  050720 000137 051364                    JMP     C.RTRN             ;RESTORE REGISTERS
9564
9565  050724 052737 100000 001462   9$:       BIS     #E.MDS,E.CONT      ;SET MULTIPLE DRIVE SELECT
9566  050732 004777 130522                    JSR     PC,@A.CONT         ;INDICATE CONTROLLER ERROR
9567  050736 000137 051364                    JMP     C.RTRN
9568
9569  050742 122765 000140 000001   10$:      CMPB    #RELEAS,P.CMND(R5) ;CHECK IF RELEASE COMMAND
9570  050750 001031                           BNE     13$                ;NO, CHECK IF READ ALL HEADERS
9571  050752 010537 001464                    MOV     R5,O.WAIT          ;STORE PARAMETER BLOCK ADDRESS IN
9572                                                                     ;  WAIT FOR COMMAND
9573  050756 052765 000010 000020             BIS     #RLS,P.CS2(R5)     ;SET RELEASE BIT
9574  050764 016562 000020 000010             MOV     P.CS2(R5),RKCS2(R2) ;LOAD CS2 FOR DESELECT
9575  050772 112737 000001 001506             MOVB    #1,I.ISRL          ;SET FLAG FOR RELEASE COMMAND
9576  051000 116565 000007 000017             MOVB    P.CS1H(R5),P.CS1+1(R5);STORE BITS 8-15 OF CS1
9577  051006 042765 165777 000016             BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT FORMAT
9578                                                                     ;  AND DRIVE TYPE
9579  051014 112765 000101 000016             MOVB    #SELDRV,P.CS1(R5)  ;STORE COMMAND
9580  051022 016562 000016 000000   11$:       MOV     P.CS1(R5),RKCS1(R2) ;ISSUE COMMAND
9581  051030 000137 051364                    JMP     C.RTRN             ;RESTORE REGISTERS
9582
9583  051034 122765 000164 000001   13$:      CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
9584  051042 001047                           BNE     30$                ;NO, CHECK IF CONTROLLER CLEAR
9585  051044 010537 001464                    MOV     R5,O.WAIT          ;SET WAITING FOR COMMAND COMMPLETION
9586  051050 016537 000010 001502             MOV     P.BALO(R5),HDR.AD  ;LOAD HEADER ADDRESS
9587  051056 132765 000020 000007             BITB    #8.CFMT,P.CS1H(R5) ;CHECK IF 22 SECTOR FORMANT
9588  051064 001404                           BEQ     14$                ;YES, LOAD 22 IN HEADER COUNT
9589  051066 012737 000024 001504             MOV     #20.,HDR.CT        ;LOAD 20 IN SECTOR COUNT
9590  051074 000403                           BR      22$                ;GO ISSUE READ HEADER COMMAND
9591
9592  051076 012737 000026 001504   14$:      MOV     #22.,HDR.CT        ;LOAD 22 IN SECTOR COUNT
9593  051104 016562 000002 000020   22$:      MOV     P.CYLN(R5),RKDCYL(R2) ;LOAD CYLINDER ADDRESS
9594  051112 016562 000004 000006             MOV     P.SECT(R5),RKDA(R2) ;LOAD TRACK NUMBER
9595  051120 016562 000020 000010             MOV     P.CS2(R5),RKCS2(R2) ;LOAD DRIVE NUMBER
9596  051126 116565 000007 000017             MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
9597  051134 042765 165777 000016             BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR ALL BITS EXCEPT DRIVE TYPE
9598                                                                     ;  AND FORMAT
9599  051142 112765 000125 000016             MOVB    #RDHEAD,P.CS1(R5) ;STORE READ HEADER COMMAND
```

```
 9600  051150  016562  000016  000000          MOV     P.CS1(R5),RKCS1(R2) ;ISSUE READ HEADER
 9601  051156  000137  051364                  JMP     C.RTRN              ;RESTORE REGISTERS
 9602
 9603  051162  122765  000176  000001  30$:    CMPB    #CONCLR,P.CMND(R5) ;CHECK IF CONTROLLER CLEAR
 9604  051170  001004                          BNE     32$                 ;NO, CHECK IF SUBSYSTEM CLEAR
 9605  051172  004037  046712                  JSR     R0,I.CCLR           ;CLEAR CONTROLLER
 9606  051176  051364                          C.RTRN                      ;ERROR RETURN
 9607  051200  000467                          BR      40$                 ;INDICATE NORMAL RETURN
 9608
 9609  051202  122765  000177  000001  32$:    CMPB    #SUBCLR,P.CMND(R5) ;CHECK IF SUBSYSTEM CLEAR
 9610  051210  001406                          BEQ     36$                 ;YES, CLEAR SUBSYSTEM
 9611  051212  052737  000100  001462  34$:    BIS     #E.ILLD,E.CONT      ;SET ILLEGAL DRIVER COMMAND
 9612  051220  004777  130234                  JSR     PC,@A.CONT          ;REPORT ERROR
 9613  051224  000457                          BR      C.RTRN              ;RESTORE REGISTERS
 9614
 9615  051226  012762  000040  000010  36$:    MOV     #SCLR,RKCS2(R2) ;ISSUE SUBSYSTEM CLEAR
 9616  051234  016265  000000  000016          MOV     RKCS1(R2),P.CS1(R5) ;STORE COMMAND AND STATUS REGISTER 1
 9617  051242  032765  100000  000016          BIT     #CERR,P.CS1(R5) ;CLEAR IF CONTROLLER ERROR RESET
 9618  051250  001406                          BEQ     37$                 ;NO, FINISH COMMAND
 9619  051252  052737  000001  001462          BIS     #BIT0,E.CONT        ;SET CLEAR SUBSYSTEM DID NOT CLEAR
 9620                                                                      ;  CONTROLLER ERROR
 9621  051260  004777  130174                  JSR     PC,@A.CONT          ;REPORT ERROR
 9622  051264  000437                          BR      C.RTRN              ;RESTORE REGISTERS
 9623
 9624  051266  013746  001470          37$:    MOV     W.MILI,-(SP)        ;LOAD 16 MILI-SECOND COUNT FOR ATTENTION
 9625                                                                      ;  TO DISAPPEAR
 9626  051272  016265  000000  000016  38$:    MOV     RKCS1(R2),P.CS1(R5) ;STORE CS1
 9627  051300  032765  040000  000016          BIT     #DI,P.CS1(R5)       ;CHECK IF ATTENTIONS CLEARED
 9628  051306  001411                          BEQ     39$                 ;YES, FINISH COMMAND
 9629  051310  005316                          DEC     (SP)                ;DECREMENT 16 MILISECOND COUNT
 9630  051312  001367                          BNE     38$                 ;CHECK DRIVE INTERRUPT AGAIN
 9631  051314  005726                          TST     (SP)+               ;ADJUST STACK
 9632  051316  052737  000040  001462          BIS     #E.SCLR,E.CONT      ;SET SUBSYSTEM CLEAR DID NOT CLEAR
 9633                                                                      ;  DRIVE ATTENTIONS
 9634  051324  004777  130130                  JSR     PC,@A.CONT          ;REPORT ERROR
 9635  051330  000415                          BR      C.RTRN              ;RESTORE REGISTER
 9636
 9637  051332  005726                  39$:    TST     (SP)+               ;ADJUST STACK
 9638  051334  032765  000400  000014          BIT     #NOCHK,P.PRST(R5) ;CHECK IF NO CHECK MODE
 9639  051342  001010                          BNE     C.RTRN              ;YES, RESTORE REGISTERS
 9640  051344  112737  177777  001506          MOVB    #-1,I.ISRL          ;SET INTERRUPT ENABLE SET
 9641  051352  012762  000100  000000          MOV     #IE,RKCS1(R2)       ;SET INTERRUPT ENABLE
 9642  051360  004777  130070                  JSR     PC,@A.NORM          ;INDICATE NORMAL TERMINATION
 9643
 9644  051364  012637  177776          C.RTRN: MOV     (SP)+,PS            ;RESTORE PSW
 9645  051370  012600                          MOV     (SP)+,R0            ;RESTORE R0
 9646  051372  012601                          MOV     (SP)+,R1            ;RESTORE R1
 9647  051374  012602                          MOV     (SP)+,R2            ;RESTORE R2
 9648  051376  012603                          MOV     (SP)+,R3            ;RESTORE R3
 9649  051400  012604                          MOV     (SP)+,R4            ;RESTORE R4
 9650  051402  012605                          MOV     (SP)+,R5            ;RESTORE R5
 9651  051404  000207                          RTS     PC                  ;RETURN
```

```
 9652                                   .SBTTL   *ENQUEUE ROUTINE
 9653
 9654                                   ;;*****************************************************************
 9655                                   ;*
 9656                                   ;*      THIS ROUTINE WILL BUT THE ADDRESS OF THE PARAMETER IN THE
 9657                                   ;*      QUEUE SPECIFIED BY THE LOCATION FOLLOWING THE CALL.
 9658                                   ;*
 9659                                   ;*CALL  JSR     R0,Q.PUSH
 9660                                   ;*      <ADDRESS OF QUEUE>
 9661                                   ;*      RETURN
 9662                                   ;*
 9663                                   ;;*****************************************************************
 9664
 9665  051406  010446                   Q.PUSH: MOV     R4,-(SP)        ;STORE R4 ON STACK
 9666  051410  010346                           MOV     R3,-(SP)        ;STORE R3 ON STACK
 9667  051412  013746  177776                   MOV     PS,-(SP)        ;STORE PSW ON STACK
 9668  051416  013737  001452  177776           MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
 9669  051424  012004                           MOV     (R0)+,R4        ;STORE QUEUE ADDRESS
 9670  051426  005065  000064                   CLR     P.QLNK(R5)      ;CLEAR QUEUE LINK
 9671  051432  005724                           TST     (R4)+           ;CHECK IF QUEUE IS EMPTY
 9672  051434  001405                           BEQ     1$              ;YES, PUT PARAMETER BLOCK ADDRESS
 9673                                                                   ;  AT HEAD AND TAIL OF THE QUEUE
 9674  051436  011403                           MOV     (R4),R3         ;STORE TAIL OF QUEUE
 9675  051440  010563  000064                   MOV     R5,P.QLNK(R3)   ;UPDATE QUEUE LINK
 9676  051444  010514                           MOV     R5,(R4)         ;UPDATE QUEUE TAIL
 9677  051446  000402                           BR      2$
 9678
 9679  051450  010514                   1$:     MOV     R5,(R4)         ;LOAD TAIL OF QUEUE
 9680  051452  010544                           MOV     R5,-(R4)        ;LOAD HEAD OF QUEUE
 9681
 9682  051454  012637  177776           2$:     MOV     (SP)+,PS        ;RESTORE PSW
 9683  051460  012603                           MOV     (SP)+,R3        ;RESTORE R3
 9684  051462  012604                           MOV     (SP)+,R4        ;RESTORE R4
 9685  051464  000200                           RTS     R0              ;RETURN
```

```
                                .SBTTL   *POP QUEUE ROUTINE

                                ;;*****************************************************************
                                ;;*
                                ;;*      THIS ROUTINE WILL PUT THE ADDRESS OF THE FIRST PARAMTER
                                ;;*      BLOCK ADDRESS OF THIS QUEUE ON THE STACK AND REMOVE IT
                                ;;*      FROM THE QUEUE DESIGNATED BY THE CALL.  A RETURN OF ZERO
                                ;;*      ON THE STACK INDICATES THAT THE QUEUE WAS EMPTY.  OTHERWISE,
                                ;;*      THE STACK WILL CONTAIN THE ADDRESS OF THE FIRST
                                ;;*      PARAMETER BLOCK ON THE QUEUE.
                                ;;*
                                ;;*CALL  JSR     R0,Q.POP
                                ;;*      <ADDRESS OF QUEUE>
                                ;;*      RETURN
                                ;;*
                                ;;*****************************************************************

051466  011646                  Q.POP:  MOV     (SP),-(SP)      ;CREATE SPACE ON THE STACK
051470  010546                          MOV     R5,-(SP)        ;STORE R5 ON THE STACK
051472  010446                          MOV     R4,-(SP)        ;STORE R4 ON THE STACK
051474  013746  177776                  MOV     PS,-(SP)        ;STORE PSW ON THE STACK
051500  013737  001452  177776          MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
051506  012004                          MOV     (R0)+,R4        ;STORE QUEUE ADDRESS
051510  011405                          MOV     (R4),R5         ;STORE PARAMETER BLOCK ADDRESS
051512  010566  000010                  MOV     R5,10(SP)       ;LOAD ADDRESS ON STACK
051516  001404                          BEQ     1$              ;IF QUEUE IS EMPTY, RETURN
051520  016524  000064                  MOV     P.QLNK(R5),(R4)+ ;READJUST HEAD OF THE QUEUE
051524  001001                          BNE     1$              ;IF NOT LAST ELEMENT OF THE QUEUE, RETURN
051526  005014                          CLR     (R4)            ;CLEAR TAIL POINTER IF LAST ELEMENT OF QUEUE

051530  012637  177776          1$:     MOV     (SP)+,PS        ;RESTORE PSW
051534  012604                          MOV     (SP)+,R4        ;RESTORE R4
051536  012605                          MOV     (SP)+,R5        ;RESTORE R5
051540  000200                          RTS     R0              ;RETURN
```

```
                                .SBTTL   *SEARCH QUEUE FOR DRIVE NOT POSITIONING

                                ;;*****************************************************************
                                ;;*
                                ;;*      THIS ROUTINE WILL SEARCH THROUGH THE COMMAND INITIATION
                                ;;*      QUEUE FOR THE FIRST PARAMETER BLOCK WHOSE DRIVE IS
                                ;;*      NOT POSITIONING.  IF THE QUEUE IS EMPTY OF ALL DRIVES
                                ;;*      ARE POSITIONING ZERO IS PLACED IN R5.  OTHERWISE,
                                ;;*      THE ADDRESS OF THE PARAMETER BLOCK IS PLACED IN R5.
                                ;;*
                                ;;*CALL  JSR     PC,Q.SRCH
                                ;;*      RETURN
                                ;;*
                                ;;*NOTE: THIS ROUTINE DESTROYS R4.
                                ;;*
                                ;;*****************************************************************

051542  012704  001324        Q.SRCH: MOV     #CINITQ,R4          ;LOAD ADDRESS OF COMMAND INITIATION
                                                                ;  QUEUE IN R4
051546  011405                1$:     MOV     (R4),R5             ;GET NEXT QUEUE ELEMENT
051550  001425                        BEQ     5$                  ;IF END OF QUEUE, RETURN
051552  032765  000002  000014        BIT     #DRVPOS,P.PRST(R5) ;CHECK IF DRIVE IS POSITIONING
051560  001404                        BEQ     2$                  ;NO, RETURN PARAMETER BLOCK ADDRESS
051562  010504                        MOV     R5,R4               ;UPDATE LINK ADDRESS POINTER
051564  062704  000064                ADD     #P.QLNK,R4          ;CALCULATE LINK ADDRESS
051570  000766                        BR      1$                  ;GET NEXT ELEMENT OF QUEUE

051572  016514  000064        2$:     MOV     P.QLNK(R5),(R4)     ;UPDATE QUEUE LINK
051576  001012                        BNE     5$                  ;CHECK IF LAST ELEMENT OF QUEUE
                                                                ;  IF NOT, DO NOT UPDATE QUEUE TAIL
051600  022704  001324                CMP     #CINITQ,R4          ;CHECK IF ONLY ELEMENT IN QUEUE
051604  001405                        BEQ     4$                  ;YES, CLEAR HEAD AND TAIL
051606  162704  000064                SUB     #P.QLNK,R4          ;ADJUST R4 FOR TAIL POINTER
051612  010437  001326                MOV     R4,CINITQ+2         ;LOAD TAIL
051616  000207                        RTS     PC                  ;RETURN

051620  005024                4$:     CLR     (R4)+               ;CLEAR HEAD OF QUEUE
051622  005014                        CLR     (R4)                ;CLEAR TAIL OF QUEUE
051624  000207                5$:     RTS     PC                  ;RETURN
```

```
                                .SBTTL   *REMOVE PARAMETER BLOCK FROM COMMAND INITIATION QUEUE
                                ;;*****************************************************************
                                ;*
                                ;*      THIS ROUTINE WILL CHECK IF THE CURRENT PARAMETER BLOCK
                                ;*      IS IN THE COMMAND INITIATION QUEUE.  IF IT IS, THIS
                                ;*      ROUTINE WILL REMOVE IT FROM THE COMMAND INITIATION QUEUE.
                                ;*
                                ;*CALL  JSR     PC,Q.RMOV
                                ;*      RETURN
                                ;*
                                ;*NOTE: R5 CONTAINS ADDRESS OF CURRENT PARAMETER BLOCK.
                                ;*
                                ;;*****************************************************************
051626  032765  040000  000014  Q.RMOV: BIT     #Q.INIT,P.PRST(R5) ;CHECK IF PARAMETER BLICK ENQUEUED IN
                                                                   ;  INITIATION QUEUE
051634  001001                        BNE     5$                 ;YES, REMOVE PARAMETER BLOCK
                                                                   ;  FROM COMMAND INITIATION QUEUE
051636  000207                        RTS     PC                 ;RETURN

051640  010446                5$:     MOV     R4,-(SP)           ;STORE R4 ON STACK
051642  010346                        MOV     R3,-(SP)           ;STORE R3 ON THE STACK
051644  012704  001324                MOV     #CINITQ,R4         ;LOAD ADDRESS OF COMMAND INITIATION
                                                                   ;  QUEUE IN R4
051650  011403                10$:    MOV     (R4),R3            ;GET NEXT QUEUE ELEMENT
051652  001001                        BNE     11$                ;CHECK IF NO MORE ELEMENTS
051654  000000                        HALT                       ;  *** FLAGS DO NOT AGREE WITH
                                                                   ;  COMMAND INITIATION QUEUE
051656  020305                11$:    CMP     R3,R5              ;CHECK IF CORRECT PARAMETER BLOCK
051660  001404                        BEQ     12$                ;YES, ADJUST QUEUE
051662  010304                        MOV     R3,R4              ;UPDATE LINK ADDRESS POINTER
051664  062704  000064                ADD     #P.QLNK,R4         ;CALCULATE LINK ADDRESS
051670  000767                        BR      10$                ;GET NEXT ELEMENT OF QUEUE

051672  016314  000064        12$:    MOV     P.QLNK(R3),(R4)    ;UPDATE QUEUE LINK
051676  001012                        BNE     15$                ;CHECK IF LAST ELEMENT OF QUEUE
                                                                   ;  IF NOT, DO NOT UPDATE QUEUE TAIL
051700  022704  001324                CMP     #CINITQ,R4         ;CHECK IF ONLY ELEMENT IN QUEUE
051704  001405                        BEQ     14$                ;YES, CLEAR HEAD AND TAIL
051706  162704  000064                SUB     #P.QLNK,R4         ;ADJUST R4 FOR TAIL POINTER
051712  010437  001326                MOV     R4,CINITQ+2        ;LOAD TAIL
051716  000402                        BR      15$                ;RETURN

051720  005024                14$:    CLR     (R4)+              ;CLEAR HEAD OF QUEUE
051722  005014                        CLR     (R4)               ;CLEAR TAIL OF QUEUE
051724  042765  040006  000014  15$:  BIC     #DRVPDT!DRVPOS!Q.INIT,P.PRST(R5) ;CLEAR DRIVE POSITIONING,
                                                                   ;  DRIVE POSITIONING FOR DATA TRANSFER, AND
                                                                   ;  PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE

051732  012603                        MOV     (SP)+,R3           ;RESTORE R3
051734  012604                        MOV     (SP)+,R4           ;RESTORE R4
051736  000207                        RTS     PC                 ;RETURN
```

```
                        .SBTTL   *COMMAND OPTIMIZER

                        ;;*****************************************************************
                        ;*
                        ;*      THIS ROUTINE WILL INITIATE THE COMMAND AS SPECIFIED IN THE
                        ;*      PARAMETER BLOCK OF THE FIRST DRIVE WHICH IS NOT POSITIONING
                        ;*      IN THE COMMAND INITIATION QUEUE.
                        ;*
                        ;*CALL  JSR     PC,C.OPT
                        ;*      RETURN
                        ;*
                        ;*      ROUTINES USED:
                        ;*              C.INIT
                        ;*              Q.PUSH
                        ;*              Q.SRCH
                        ;*
                        ;;*****************************************************************

                        C.OPT:  MOV     R5,-(SP)        ;STORE R5 ON STACK
                                MOV     R4,-(SP)        ;STORE R4 ON STACK
                                MOV     R3,-(SP)        ;STORE R3 ON STACK
                                MOV     R2,-(SP)        ;STORE R2 ON STACK
                                MOV     R1,-(SP)        ;STORE R1 ON STACK
                                MOV     R0,-(SP)        ;STORE R0 ON STACK
                                MOV     PS,-(SP)        ;STORE PSW ON STACK
                                MOV     #1,OPTFLG       ;SET FLAG FOR CMND OPTIMIZER
                                MOV     RKPRI,PS        ;LOCK OUT RK06 INTERRUPTS
                                TST     O.WAIT          ;CHECK IF WAITING FOR COMMAND COMPLETION
                                BNE     12$             ;YES, RESTORE REGISTERS
                                MOV     RKBAS,R2        ;LOAD R2 WITH BASE ADDRESS OF THE RK06
                                                        ;  REGISTERS
                                MOV     RKCS1(R2),T.CS1 ;STORE COMMAND AND STATUS REGISTER 1
                                BIT     #RDY,T.CS1      ;CHECK IF READY SET
                                BEQ     12$             ;NO, WAIT FOR COMMAND COMPLETION
                                JSR     PC,Q.SRCH       ;SEARCH FOR PARAMETER BLOCK ADDRESS
                                TST     R5              ;CHECK IF NO COMMAND CAN BE ISSUED
                                BNE     1$              ;YES, ISSUE COMMAND
                                BIT     #DI,T.CS1       ;CHECK IF DRIVE INTERRUPT WAITING
                                BEQ     12$             ;NO, RETURN
                                MOVB    #-1,I.ISRL      ;SET FLAG THAT INTERRUPT WAS FORCED
                                MOV     #INTR,RKCS1(R2) ;FORCE INTERRUPT TO SCAN DRIVE ATTENTIONS
                        12$:    JMP     O.RTRN          ;RESTORE REGISTERS

                        1$:     MOVB    #DR.SEL,T.CS1   ;SET DRIVE SELECT BIT
                                JSR     R0,I.ISSU       ;ISSUE DRIVE SELECT COMMAND
                                O.RTRN                  ;ERROR RETURN ADDRESS
                                BIT     #NED,T.CS2      ;SEE IF NED ERROR SET
                                BNE     24$             ;BR IF YES
                                BIT     #DRA,T.DS       ;SEE IF DRIVE IS AVAILABLE
                                BNE     24$             ;BR IF AVAILABLE
                                MOV     T.CS2,R3        ;GET RKCS2
                                BIC     #177770,R3      ;CLEAR ALL BUT DRIVE NO.
                                BISB    INTMSK(R3),W.TIME ;SET BIT FOR THIS DRIVE
                                ASL     R3
                                MOV     W.MIN,W.DRV(R3) ;SET TIMER FOR 1 MINUTE FOR RELEASE
                                CLR     O.WAIT          ;CLEAR WAITING FOR CMND COMPLETION
```

```
9868  052152  004037  046712                  JSR     R0,I.CCLR       ;ISSUE CONTROLLER CLEAR
9869  052156  052444                          0.RTRN                  ;ERROR RETURN ADDRESS
9870  052160  042765  C40006  000014          BIC     #DRVPOS!DRVPDT!Q.INIT,P.PRST(R5)
9871  052166  052765  010000  000014  10$:    BIS     #DRVSZD,P.PRST(R5) ;SET DRIVE SIEZED FLAG
9872  052174  000137  052444                  JMP     0.RTRN          ;EXIT
9873  052200                          24$:
9874  052200  005065  000212                  CLR     P.ERR(R5)       ;CLEAR ERROR STATUS INFORMATION
9875  052204  122765  000164  000001          CMPB    #RDALHD,P.CMND(R5) ;CHECK IF READ ALL HEADERS
9876  052212  001410                          BEQ     2$              ;YES, CHECK IF IMPLIED SEEK
9877  052214  122765  000121  000001          CMPB    #RDDATA,P.CMND(R5) ;CHECK IF DATA TRANSFER
9878  052222  101021                          BHI     3$              ;NO, GO TO THE COMMAND INITIATOR
9879  052224  122765  000131  000001          CMPB    #WRTCHK,P.CMND(R5) ;CHECK IF SPECIAL COMMAND
9880  052232  103415                          BLO     3$              ;YES, GO TO THE COMMAND INITIATOR
9881  052234  116504  000000          2$:     MOVB    P.DRVN(R5),R4   ;STORE DRIVE NUMBER
9882  052240  136437  001514  001513          BITB    INTMSK(R4),0.OVER ;CHECK IF OVERLAPPED OVERATIONS FOR THIS DRIVE
9883  052246  001407                          BEQ     3$              ;NO, GO TO THE COMMAND INITIATOR
9884  052250  032765  000004  000014          BIT     #DRVPDT,P.PRST(R5) ;CHECK IF POSITIONING FOR
9885                                                                  ;  DATA TRANSFER OCCURED
9886  052256  001411                          BEQ     5$              ;NO, ISSUE SEEK
9887  052260  042765  040004  000014          BIC     #DRVPDT!Q.INIT,P.PRST(R5) ;CLEAR POSITIONING IN PROGRESS FOR
9888                                                                  ;  DATA TRANSFER AND PARAMETER BLOCK ENQUEUED
9889                                                                  ;  IN INITIATION QUEUE
9890
9891  052266  010537  052276          3$:     MOV     R5,4$           ;LOAD PARAMETER BLOCK ADDRESS
9892  052272  004737  047716                  JSR     PC,C.INIT       ;ISSUE COMMAND
9893  052276  000000                  4$:     .WORD   0               ;ADDRESS OF PARAMETER BLOCK
9894                                                                  ;  PLACED HERE
9895  052300  000461                          BR      0.RTRN          ;RESTORE REGISTERS
9896
9897  052302  156437  001514  001512  5$:     BISB    INTMSK(R4),W.TIME ;SET WATCH DOG TIMING
9898  052310  010403                          MOV     R4,R3           ;STORE DRIVE NUMBER
9899  052312  006303                          ASL     R3              ;MULTIPLY DRIVE NUMBER BY 2
9900  052314  013763  001472  001544          MOV     W.SEC,W.DRV(R3) ;LOAD WATCH-DOG TIME
9901  052322  005037  001464                  CLR     0.WAIT          ;CLEAR WAIT FOR COMMAND
9902  052326  105037  001506                  CLRB    I.ISRL          ;CLEAR RELEASE OF INTERRUPT ISSUED
9903
9904                                  ;       RESET ALL THE BITS IN THE PROGRAM STATUS REGISTER EXCEPT
9905                                  ;;              DRIVE IN USE
9906                                  ;;              WRITE FOR WRITE CHECK
9907                                  ;;              NO CHECK
9908                                  ;;              DROP DRIVE FROM TEST SEQUENCE
9909                                  ;;              INHIBIT BUS ADDRESS INCREMENT
9910  052332  042765  075176  000014          BIC     #↑C<DRVUSE!W.WCK!NOCHK!DRPDRV!DTBAII>,P.PRST(R5)
9911
9912                                  ;       SET DRIVE POSITIONING, DRIVE POSITIONING BECAUSE OF DATA TRANSFER,
9913                                  ;         AND PARAMETER BLOCK ENQUEUED IN INITIATION QUEUE
9914  052340  052765  040006  000014          BIS     #DRVPOS!DRVPDT!Q.INIT,P.PRST(R5)
9915
9916  052346  010500                          MOV     R5,R0           ;STORE PARAMTER BLOCK ADDRESS
9917  052350  062700  000016                  ADD     #P.CS1,R0       ;CALCULATE FIRST LOCATION TO BE CLEARED
9918  052354  010501                          MOV     R5,R1           ;STORE PARAMETER BLOCK ADDRESS
9919  052356  062701  000062                  ADD     #P.EPAT,R1      ;CALCULATE LAST LOCATION TO BE CLEARED
9920
9921  052362  005020                  6$:     CLR     (R0)+           ;CLEAR RETURN PARAMETERS
9922  052364  020001                          CMP     R0,R1           ;CHECK IF FINISHED
9923  052366  101775                          BLOS    6$              ;NO, CLEAR NEXT RETURN PARAMETER
```

```
 9924  052370  005062  000026                    CLR     RKMR1(R2)              ;CLEAR MAINTENANCE REGISTER 1
 9925  052374  016562  000002  000020            MOV     P.CYLN(R5),RKDCYL(R2)  ;LOAD CYLINDER ADDRESS
 9926  052402  010462  000010                    MOV     R4,RKCS2(R2)           ;LOAD DEVICE NUMBER
 9927  052406  116565  000007  000017            MOVB    P.CS1H(R5),P.CS1+1(R5) ;STORE BITS 8-15 OF CS1
 9928  052414  042765  165777  000016            BIC     #^C<CFMT!CDT>,P.CS1(R5) ;CLEAR BITS EXCEPT FORMAT AND
 9929                                                                          ;  DRIVE TYPE
 9930  052422  112765  000117  000016            MOVB    #SEEK,P.CS1(R5)        ;STORE COMMAND
 9931  052430  016562  000016  000000            MOV     P.CS1(R5),RKCS1(R2)    ;ISSUE SEEK
 9932  052436  004037  051406                    JSR     R0,Q.PUSH              ;REQUEUE COMMAND IN COMMAND
 9933  052442  001324                            CINITQ                         ;  INITIATION QUEUE
 9934
 9935  052444  005037  001500          O.RTRN:   CLR     OPTFLG                 ;CLEAR OPTIMIZER FLAG
 9936  052450  012637  177776                    MOV     (SP)+,PS               ;RESTORE PSW
 9937  052454  012600                            MOV     (SP)+,R0               ;RESTORE R0
 9938  052456  012601                            MOV     (SP)+,R1               ;RESTORE R1
 9939  052460  012602                            MOV     (SP)+,R2               ;RESTORE R2
 9940  052462  012603                            MOV     (SP)+,R3               ;RESTORE R3
 9941  052464  012604                            MOV     (SP)+,R4               ;RESTORE R4
 9942  052466  012605                            MOV     (SP)+,R5               ;RESTORE R5
 9943  052470  000207                            RTS     PC                     ;RETURN
```

```
                                        .SBTTL  MEMORY CHECK ENABLE TEST
                                        
052472  104401  064700          MEMERR: TYPE    ,ERR400         ;TYPE UNEXPECTED MEMORY PARITY TRAP
052476  011646                          MOV     (SP),-(SP)      ;TYPE PC VALUE
052500  004737  052774                  JSR     PC,BINOCT
052504  104401  001165                  TYPE    ,$CRLF
052510  000002                          RTI                     ;RETURN
```

```
 9951                                   .SBTTL  TYPE ROUTINE
 9952
 9953                                   ;;**********************************************************************
 9954                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 9955                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 9956                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 9957                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 9958                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 9959                                   ;*
 9960                                   ;*CALL:
 9961                                   ;*1) USING A TRAP INSTRUCTION
 9962                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 9963                                   ;*OR
 9964                                   ;*      TYPE
 9965                                   ;*      MESADR
 9966                                   ;*
 9967
 9968  052512  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 9969  052516  100002                          BPL     1$              ;;BR IF YES
 9970  052520  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
 9971  052522  000430                          BR      3$              ;;LEAVE
 9972  052524  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
 9973  052526  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 9974  052532  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 9975  052540  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 9976  052542  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 9977  052550  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 9978  052552  010037  052562                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 9979  052556  004737  055424                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 9980  052562  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
 9981  052564  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 9982  052572  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
 9983  052574  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 9984  052576  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 9985  052600  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 9986  052602  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
 9987  052604  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
 9988  052610  000002                          RTI                     ;;RETURN
 9989  052612  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 9990  052616  001430                          BEQ     8$
 9991  052620  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 9992  052624  001006                          BNE     5$
 9993  052626  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
 9994  052630  104401                          TYPE                    ;;TYPE A CR AND LF
 9995  052632  001165                          $CRLF
 9996  052634  105037  052770                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
 9997  052640  000755                          BR      2$              ;;GET NEXT CHARACTER
 9998  052642  004737  052724          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
 9999  052646  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
10000  052652  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
10001  052654  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
10002                                                                  ;;AND THE NULL CHAR.
10003  052660  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
10004  052664  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
10005  052666  004737  052724                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
10006  052672  105337  052770                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
10007  052676  000770                        BR      7$              ;;LOOP
10008
10009                                  ;HORIZONTAL TAB PROCESSOR
10010
10011  052700  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
10012  052704  004737  052724          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
10013  052710  132737  000007  052770          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
10014  052716  001372                          BNE     9$              ;;TAB STOP
10015  052720  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
10016  052722  000724                          BR      2$              ;;GET NEXT CHARACTER
10017  052724  105777  126220          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
10018  052730  100375                          BPL     $TYPEC
10019  052732  116677  000002  126212        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
10020  052740  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
10021  052746  001003                          BNE     1$              ;;BRANCH IF NO
10022  052750  105037  052770                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
10023  052754  000406                          BR      $TYPEX          ;;EXIT
10024  052756  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
10025  052764  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
10026  052766  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
10027  052770  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
10028  052772  000207          $TYPEX: RTS     PC
10029
10030
10031                                  .SBTTL  BINARY TO OCTAL (ASCII) CONVERSION
10032
10033                                  ;;*****************************************************************
10034                                  ;*
10035                                  ;*      THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A
10036                                  ;*      6-DIGIT OCTAL (ASCII) NUMBER.
10037                                  ;*
10038                                  ;*CALL  MOV     NUM,-(SP)
10039                                  ;*      JSR     PC,BINOCT
10040                                  ;*      RETURN                  ;NUMBER IN OCTSTG
10041                                  ;*
10042                                  ;;*****************************************************************
10043
10044  052774                          BINOCT:
10045  052774  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10046  052776  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10047  053000  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10048  053002  012704  000006                  MOV     #6,R4           ;STORE COUNT
10049  053006  016605  000010                  MOV     10(SP),R5       ;PICKUP INPUT NUMBER
10050  053012  012703  053070                  MOV     #OCTSTG,R3      ;LOAD ADDRESS OF CONVERTTED STRING
10051  053016  005013                          CLR     (R3)            ;CLEAR OUTPUT BYTE
10052  053020  006105                          ROL     R5              ;ROTATE MSB INTO "C"
10053  053022  000404                          BR      3$              ;GO DO MSB
10054
10055  053024  006105                  2$:     ROL     R5              ;FORM THIS DIGIT
10056  053026  006105                          ROL     R5
10057  053030  006105                          ROL     R5
10058  053032  110513                          MOVB    R5,(R3)
10059  053034  106113                  3$:     ROLB    (R3)            ;GET LSB OF THIS DIGIT
10060  053036  142713  177770                BICB    #177770,(R3)    ;GET RID OF JUNK
10061  053042  152723  000060                BISB    #'0,(R3)+       ;MAKE IT ASCII
10062  053046  005304                          DEC     R4              ;CHECK IF DONE
```

```
10063  053050  001365                        BNE     2$              ;NO CONVERT NEXT DIGIT
10064  053052  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
10065  053054  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
10066  053056  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
10067  053060  012616                        MOV     (SP)+,(SP)      ;SETUP STACK FOR RETURN
10068  053062  104401  053070                TYPE    ,OCTSTG         ;TYPE NUMBER
10069  053066  000207                        RTS     PC              ;RETURN
10070
10071  053070  000006                OCTSTG: .BLKB   6
10072  053076     000      000             .BYTE   0,0
10073
10074                                  .SBTTL  TTY INPUT ROUTINE
10075
10076                                  ;;******************************************************************
10077                                  .ENABL  LSB
10078
10079                                  .DSABL  LSB
10080
10081
10082                                  ;;******************************************************************
10083                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
10084                                  ;*CALL:
10085                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
10086                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
10087                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
10088                                  ;
10089
10090  053100  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
10091  053102  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
10092  053110  105777  126030        1$:     TSTB    @$TKS           ;;WAIT FOR
10093  053114  100375                        BPL     1$              ;;A CHARACTER
10094  053116  117766  126024  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
10095  053124  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
10096  053132  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
10097  053140  001013                        BNE     3$              ;;BRANCH IF NO
10098  053142  105777  125776        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
10099  053146  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
10100  053150  117746  125772                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
10101  053154  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10102  053160  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
10103  053164  001366                        BNE     2$              ;;IF NOT DISCARD IT
10104  053166  000750                        BR      1$              ;;YES, RESUME
10105  053170  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
10106  053176  002407                        BLT     4$              ;;BRANCH IF YES
10107  053200  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
10108  053206  003003                        BGT     4$              ;;BRANCH IF YES
10109  053210  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
10110  053216  000002                4$:     RTI                     ;;GO BACK TO USER
10111                                  ;;******************************************************************
10112                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
10113                                  ;*CALL:
10114                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
10115                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
10116                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
10117
10118  053220  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
```

```
10119  053222  005046                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
10120  053224  012703  053454          1$:   MOV     #$TTYIN,R3      ;;GET ADDRESS
10121  053230  022703  053505          2$:   CMP     #$TTYIN+25.,R3  ;;BUFFER FULL?
10122  053234  101456                        BLOS    4$              ;;BR IF YES
10123  053236  104402                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
10124  053240  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
10125  053242  122713  000177          10$:  CMPB    #177,(R3)       ;;IS IT A RUBOUT
10126  053246  001022                        BNE     5$              ;;BR IF NO
10127  053250  005716                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
10128  053252  001007                        BNE     6$              ;;BR IF NO
10129  053254  112737  000134  053452        MOVB    #'\,9$          ;;TYPE A BACK SLASH
10130  053262  104401  053452                TYPE    9$
10131  053266  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
10132  053272  005303                  6$:   DEC     R3              ;;BACKUP BY ONE
10133  053274  020327  053454                CMP     R3,#$TTYIN      ;;STACK EMPTY?
10134  053300  103434                        BLO     4$              ;;BR IF YES
10135  053302  111337  053452                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
10136  053306  104401  053452                TYPE    9$              ;;GO TYPE
10137  053312  000746                        BR      2$              ;;GO READ ANOTHER CHAR.
10138  053314  005716                  5$:   TST     (SP)            ;;RUBOUT KEY SET?
10139  053316  001406                        BEQ     7$              ;;BR IF NO
10140  053320  112737  000134  053452        MOVB    #'\,9$          ;;TYPE A BACK SLASH
10141  053326  104401  053452                TYPE    9$
10142  053332  005016                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
10143  053334  122713  000025          7$:   CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
10144  053340  001003                        BNE     8$              ;;BR IF NO
10145  053342  104401  053505                TYPE    $CNTLU          ;;TYPE A CONTROL "U"
10146  053346  000726                        BR      1$              ;;GO START OVER
10147  053350  122713  000022          8$:   CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
10148  053354  001011                        BNE     3$              ;;BRANCH IF NO
10149  053356  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
10150  053360  104401  001165                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
10151  053364  104401  053454                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
10152  053370  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
10153  053372  104401  001164          4$:   TYPE    $QUES           ;;TYPE A '?'
10154  053376  000712                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
10155  053400  111337  053452          3$:   MOVB    (R3),9$         ;;ECHO THE CHARACTER
10156  053404  104401  053452                TYPE    9$
10157  053410  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
10158  053414  001305                        BNE     2$              ;;LOOP IF NOT RETURN
10159  053416  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
10160  053422  104401  001166                TYPE    $LF             ;;TYPE A LINE FEED
10161  053426  005726                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
10162  053430  012603                        MOV     (SP)+,R3        ;;RESTORE R3
10163  053432  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
10164  053434  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
10165  053442  012766  053454  000004        MOV     #$TTYIN,4(SP)
10166  053450  000002                        RTI                     ;;RETURN
10167  053452     000                  9$:   .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
10168  053453     000                        .BYTE   0               ;;TERMINATOR
10169  053454  000031             $TTYIN:   .BLKB   25.             ;;RESERVE 25. BYTES FOR TTY INPUT
10170  053505     136  006525  000012  $CNTLU:   .ASCIZ  /↑U/<15><12>       ;;CONTROL "U"
10171  053512  043536  005015     000  $CNTLG:   .ASCIZ  /↑G/<15><12>       ;;CONTROL "G"
10172  053517     015  051412  051127  $MSWR:    .ASCIZ  <15><12>/SWR = /
10173  053524  036440  000040
10174  053530  020040  042516  020127  $MNEW:    .ASCIZ  /  NEW = /
```

```
 10175  053536  020075  000
 10176          053542                  .EVEN
 10177
 10178                                  .SBTTL  TK INITIALIZE ROUTINE
 10179
 10180                                  ;;*****************************************************************
 10181                                  ;*
 10182                                  ;*      THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD TO ALL THE
 10183                                  ;*        OPERATOR TO COMMUNICATE WITH THE PROGRAM.
 10184                                  ;*
 10185                                  ;*CALL
 10186                                  ;*      JSR     PC,$TKINT
 10187                                  ;*      RETURN
 10188                                  ;*
 10189                                  ;;*****************************************************************
 10190
 10191  053542  012737  053572  000060  $TKINT: MOV     #$TKSRV,TKVEC   ;INITIALIZE THE KEY BOARD VECTOR
 10192  053550  013737  001452  000062          MOV     RKPRI,TKVEC+2   ;LOCK RK06 AND TTY INTERUPTS
 10193  053556  005777  125364                  TST     @$TKB           ;CLEAR DONE FLAG
 10194  053562  012777  000100  125354          MOV     #IE,@$TKS       ;ENABLE INTERRUPT
 10195  053570  000207                          RTS     PC              ;RETURN
 10196
 10197                                  .SBTTL  TK SERVICE ROUTINE (ONLY CONTROL C IS REGNIZED)
 10198
 10199  053572  117737  125350  054134  $TKSRV: MOVB    @$TKB,2$        ;PICK UP CHARACTER AND STORE FOR
 10200                                                                  ;  PRINT OUT
 10201  053600  142737  000200  054134          BICB    #BIT7,2$        ;STRIP THE JUNK
 10202  053606  122737  000003  054134          CMPB    #3,2$           ;IS IT A CONTROL-C
 10203  053614  001022                          BNE     1$              ;NO, PRINT <CHAR>?<15><12>
 10204  053616  104401  054141                  TYPE    ,$CNTLC         ;TYPE A CONTROL-C
 10205  053622  005077  125316                  CLR     @$TKS           ;CLEAR INTERRUPT ENABLE
 10206  053626  016637  000002  177776          MOV     2(SP),PS        ;REINSTATE PREVIOUS PSW
 10207  053634  004737  011614                  JSR     PC,OPRCMD       ;PROCESS OPERATOR COMMAND
 10208  053640  013737  001452  177776          MOV     RKPRI,PS        ;LOAD PSW TO LOCK OUT TTY AND RK06
 10209  053646  005777  125274                  TST     @$TKB           ;CLEAR DONE FLAG
 10210  053652  012777  000100  125264          MOV     #IE,@$TKS       ;ENABLE INTERRUPTS
 10211  053660  000002                          RTI                     ;RETURN
 10212
 10213  053662  122737  000007  054134  1$:     CMPB    #7,2$           ;CHECK IF CONTROL G
 10214  053670  001403                          BEQ     5$              ;YES, GET NEW SWITCH REGISTER VALUE
 10215  053672  104401  054134                  TYPE    ,2$             ;PRINT CHARACTER ?<15><12>
 10216  053676  000002                  3$:     RTI                     ;RETURN
 10217
 10218  053700  022737  000176  001140  5$:     CMP     #SWREG,SWR      ;CHECK IF SOFTWARE SWITCH REG
 10219  053706  001373                          BNE     3$              ;NO, RETURN
 10220  053710  122737  000001  001134          CMPB    #1,$AUTOB       ;CHECK IF IN AUTO ACCEPT MODE
 10221  053716  001767                          BEQ     3$              ;YES, RETURN
 10222  053720  104401  053512                  TYPE    ,$CNTLG         ;TYPE CONTROL G
 10223  053724  104401  053517                  TYPE    ,$MSWR          ;TYPE OLD SWITCH REGISTER
 10224  053730  013746  000176                  MOV     SWREG,-(SP)
 10225  053734  004737  052774                  JSR     PC,BINOCT
 10226  053740  104401  053530                  TYPE    ,$MNEW
 10227  053744  005077  125174                  CLR     @$TKS           ;GO INTO FLAG MODE
 10228  053750  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 10229  053752  005046                          CLR     -(SP)           ;;THE NEW SWR
 10230  053754  105777  125164          7$:     TSTB    @$TKS           ;;CHAR THERE?
```

```
10231  053760  100375                  BPL     7$              ;;IF NOT TRY AGAIN
10232
10233  053762  117746  125160          MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
10234  053766  042716  177600          BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
10235
10236  053772  021627  000025    9$:   CMP     (SP),#25        ;;IS IT A CONTROL-U?
10237  053776  001005                  BNE     10$             ;;BRANCH IF NOT
10238  054000  104401  053505          TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
10239  054004  062706  000006    20$:  ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
10240  054010  000757                  BR      19$             ;;LET'S TRY IT AGAIN
10241
10242
10243  054012  021627  000015    10$:  CMP     (SP),#15        ;;IS IT A <CR>?
10244  054016  001016                  BNE     16$             ;;BRANCH IF NO
10245  054020  005766  000004          TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
10246  054024  001403                  BEQ     11$             ;;BRANCH IF YES
10247  054026  016677  000002  125104  MOV     2(SP),@SWR      ;;SAVE NEW SWR
10248  054034  062706  000006    11$:  ADD     #6,SP           ;;CLEAR UP STACK
10249  054040  104401  001165    14$:  TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
10250  054044  012777  000100  125072  MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
10251  054052  000002    15$:  RTI             ;;RETURN
10252  054054  004737  052724    16$:  JSR     PC,$TYPEC       ;;ECHO CHAR
10253  054060  021627  000060          CMP     (SP),#60        ;;CHAR < 0?
10254  054064  002420                  BLT     18$             ;;BRANCH IF YES
10255  054066  021627  000067          CMP     (SP),#67        ;;CHAR > 7?
10256  054072  003015                  BGT     18$             ;;BRANCH IF YES
10257  054074  042726  000060          BIC     #60,(SP)+       ;;STRIP-OFF ASCII
10258  054100  005766  000002          TST     2(SP)           ;;IS THIS THE FIRST CHAR
10259  054104  001403                  BEQ     17$             ;;BRANCH IF YES
10260  054106  006316                  ASL     (SP)            ;;NO, SHIFT PRESENT
10261  054110  006316                  ASL     (SP)            ;;   CHAR OVER TO MAKE
10262  054112  006316                  ASL     (SP)            ;;   ROOM FOR NEW ONE.
10263  054114  005266  000002    17$:  INC     2(SP)           ;;KEEP COUNT OF CHAR
10264  054120  056616  177776          BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
10265  054124  000713                  BR      7$              ;;GET THE NEXT ONE
10266  054126  104401  001164    18$:  TYPE    ,$QUES          ;;TYPE ?<CR><LF>
10267  054132  000724                  BR      20$             ;;SIMULATE CONTROL-U
10268
10269  054134     000    2$:   .BYTE   0                       ;STORAGE FOR QUESTIONABLE INPUT
10270  054135     077  005015     000  .ASCIZ  /?/<15><12>
10271  054141     136  006503  000012  $CNTLC: .ASCIZ  /^C/<15><12>    ;CONTROL-C
10272
10273                      .SBTTL  OCTAL TO BINARY CONVERSION ROUTINE
10274
10275                      ;;*****************************************************************
10276                      ;*
10277                      ;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
10278                      ;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL
10279                      ;*      IT  WILL GENERATE TWO BINARY WORDS PLACING THE LOW 16 BITS
10280                      ;*      ON THE STACK AND THE HIGH 16 BITS IN LOCATION $HIOCT.
10281                      ;*
10282                      ;*CALL
10283                      ;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
10284                      ;*      JSR     PC,OCTBIN
10285                      ;*      <ADDRESS OF ERROR RETURN>
10286                      ;*      RETURN
```

```
10287                                   ;*
10288                                   ;;*****************************************************************
10289
10290  054146  010046                   OCTBIN: MOV     R0,-(SP)        ;SAVE R0
10291  054150  010146                           MOV     R1,-(SP)        ;SAVE R1
10292  054152  010246                           MOV     R2,-(SP)        ;SAVE R2
10293  054154  016600  000010                   MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
10294  054160  005001                           CLR     R1              ;CLEAR DATA WORDS
10295  054162  005002                           CLR     R2
10296  054164  112046                   2$:     MOVB    (R0)+,-(SP)     ;PICK THIS CHARACTER
10297  054166  001423                           BEQ     3$              ;IF ZERO GET OUT
10298  054170  121627  000054                   CMPB    (SP),#',        ;CHECK IF COMMA
10299  054174  001420                           BEQ     3$              ;IF COMMA GET OUT
10300  054176  122716  000060                   CMPB    #'0,(SP)        ;MAKE SURE THIS CHARACTER IS
10301  054202  003030                           BGT     4$              ;  AN OCTAL DIGIT
10302  054204  122716  000067                   CMPB    #'7,(SP)
10303  054210  002425                           BLT     4$
10304  054212  006301                           ASL     R1              ; *2
10305  054214  006102                           ROL     R2
10306  054216  006301                           ASL     R1              ; *4
10307  054220  006102                           ROL     R2
10308  054222  006301                           ASL     R1              ; *8
10309  054224  006102                           ROL     R2
10310  054226  042716  177770                   BIC     #^C7,(SP)       ;STRIP THE ASCII JUNK
10311  054232  062601                           ADD     (SP)+,R1        ;ADD THIS DIGIT
10312  054234  000753                           BR      2$              ;LOOP
10313  054236  005726                   3$:     TST     (SP)+           ;CLEAN PARTIAL FROM STACK
10314  054240  010166  000010                   MOV     R1,10(SP)       ;SAVE RESULT
10315  054244  010237  054300                   MOV     R2,$HIOCT
10316  054250  012602                           MOV     (SP)+,R2        ;RESTORE R2
10317  054252  012601                           MOV     (SP)+,R1        ;RESTORE R1
10318  054254  012600                           MOV     (SP)+,R0        ;RESTORE R0
10319  054256  062716  000002                   ADD     #2,(SP)         ;ADJUST RETURN
10320  054262  000207                           RTS     PC              ;RETURN
10321
10322  054264  005726                   4$:     TST     (SP)+           ;CLEAN UP PARTIAL FROM STACK
10323  054266  012602                           MOV     (SP)+,R2        ;RESTORE R2
10324  054270  012601                           MOV     (SP)+,R1        ;RESTORE R1
10325  054272  012600                           MOV     (SP)+,R0        ;RESTORE R0
10326  054274  013616                           MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ROUTINE ON STACK
10327  054276  000207                           RTS     PC              ;GO PROCESS ERROR
10328  054300  000000                   $HIOCT: .WORD   0               ;HIGH ORDER BITS GO HERE
10329                                   .SBTTL  DECIMAL TO BINARY CONVERSION ROUTINE
10330
10331                                   ;;*****************************************************************
10332                                   ;*
10333                                   ;*      THIS ROUTINE WILL CHECK A STRING OF ASCII CHARACTERS TERMINATED
10334                                   ;*      WITH A NULL <000> OR COMMA.  IF THE CHARACTERS ARE LEGAL,
10335                                   ;*      IT WILL GENERATE A  BINARY WORD PLACING IT ON THE STACK.
10336                                   ;*
10337                                   ;*CALL
10338                                   ;*      MOV     <ADDRESS OF ASCII STRING>,-(SP)
10339                                   ;*      JSR     PC,DECBIN
10340                                   ;*      <ADDRESS OF ERROR RETURN>
10341                                   ;*      RETURN
10342                                   ;*
```

```
10343                                   ;;*****************************************************************
10344
10345  054302  010046                   DECBIN: MOV     R0,-(SP)        ;SAVE R0
10346  054304  010146                           MOV     R1,-(SP)        ;SAVE R1
10347  054306  010246                           MOV     R2,-(SP)        ;SAVE R2
10348  054310  016600  000010                   MOV     10(SP),R0       ;GET ADDRESS OF ASCII STRING
10349  054314  005046                           CLR     -(SP)           ;CLEAR DATA WORD
10350  054316  005002                           CLR     R2              ;SIGN SET POSITIVE
10351  054320  122710  000055                   CMPB    #'-,(R0)        ;SEE IF A MINUS SIGN
10352  054324  001001                           BNE     2$              ;BRANCH IF NO MINUS SIGN
10353  054326  112002                           MOVB    (R0)+,R2        ;SAVE FOR LATER USE
10354  054330  112001                   2$:     MOVB    (R0)+,R1        ;PICKUP THIS CHARACTER
10355  054332  001427                           BEQ     3$              ;GET OUT IF ZERO
10356  054334  120127  000054                   CMPB    R1,#',          ;CHECK IF COMMA
10357  054340  001424                           BEQ     3$              ;GET OUT IF COMMA
10358  054342  122701  000060                   CMPB    #'0,R1          ;MAKE SURE THIS CHARACTER IS
10359  054346  003034                           BGT     5$              ;  A DIGIT BETWEEN 0 & 9
10360  054350  122701  000071                   CMPB    #'9,R1
10361  054354  002431                           BLT     5$
10362  054356  032716  170000                   BIT     #170000,(SP)    ;DON'T LET MUMBER GET TO BIG
10363  054362  001026                           BNE     5$              ;BRANCH IF NUMBER WOULD OVERFLOW
10364  054364  006316                           ASL     (SP)            ; *2
10365  054366  011646                           MOV     (SP),-(SP)      ;SAVE FOR LATER
10366  054370  006316                           ASL     (SP)            ; *4
10367  054372  006316                           ASL     (SP)            ; *8
10368  054374  062616                           ADD     (SP)+,(SP)      ; *10
10369  054376  102420                           BVS     5$              ;OVERFLOW ISN'T ALLOWED
10370  054400  162701  000060                   SUB     #'0,R1          ;STRIP AWAY THE ASCII JUNK
10371  054404  060116                           ADD     R1,(SP)         ;ADD IN THIS DIGIT
10372  054406  102414                           BVS     5$              ;OVERFLOW ISN'T ALLOWED
10373  054410  000747                           BR      2$              ;LOOP
10374  054412  005702                   3$:     TST     R2              ;CHECK IF NUMBER IS NEGATIVE
10375  054414  001401                           BEQ     4$              ;BRANCH IF NO
10376  054416  005416                           NEG     (SP)            ;YES--NEGATE THE NUMBER
10377  054420  012666  000010           4$:     MOV     (SP)+,10(SP)    ;SAVE RESULT
10378  054424  012602                           MOV     (SP)+,R2        ;RESTORE R2
10379  054426  012601                           MOV     (SP)+,R1        ;RESTORE R1
10380  054430  012600                           MOV     (SP)+,R0        ;RESTORE R0
10381  054432  062716  000002                   ADD     #2,(SP)         ;ADJUST RETURN
10382  054436  000207                           RTS     PC              ;RETURN
10383
10384  054440  005726                   5$:     TST     (SP)+           ;CLEAN PARTIAL NUMBER FROM STACK
10385  054442  012602                           MOV     (SP)+,R2        ;RESTORE R2
10386  054444  012601                           MOV     (SP)+,R1        ;RESTORE R1
10387  054446  012600                           MOV     (SP)+,R0        ;RESTORE R0
10388  054450  013616                           MOV     @(SP)+,(SP)     ;PUT ADDRESS OF ERROR ON STACK
10389  054452  000207                           RTS     PC              ;GO PROCESS ERROR
10390                                   .SBTTL  ROUTINE TO SIZE MEMORY
10391
10392                                   ;;*****************************************************************
10393                                   ;*CALL:
10394                                   ;*      JSR     PC,$SIZE
10395                                   ;*      RETURN
10396                                   ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
10397
10398  054454  010046                   $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
```

```
10399  054456  010146                  MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
10400  054460  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
10401  054464  013746  000006          MOV     @#ERRVEC+2,-(SP)
10402  054470  010600                  MOV     SP,R0           ;;SAVE THE STACK POINTER
10403                          ;;SET THE ERRVEC PS TO THE PRESENT PS
10404  054472  104400                  TRAP                    ;;PUSH OLD PSW AND PC ON STACK
10405  054474  012637  000006          MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
10406  054500  012737  054520  000004  MOV     #2$,@#ERRVEC    ;;SET FOR TIMEOUT
10407  054506  012701  020000          MOV     #20000,R1       ;;FIRST ADDRESS
10408  054512  005711          1$:     TST     (R1)            ;;TEST THIS ADDRESS
10409  054514  005721                  TST     (R1)+           ;;STEP TO NEXT ADDRESS
10410  054516  000775                  BR      1$              ;;TRY ANOTHER
10411  054520  162701  000002          2$:     SUB     #2,R1           ;;DROP BACK
10412  054524  010006                  MOV     R0,SP           ;;RESTORE THE STACK
10413  054526  012637  000006          MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
10414  054532  012637  000004          MOV     (SP)+,@#ERRVEC
10415  054536  010137  054550          MOV     R1,$LSTAD       ;;LAST ADDRESS
10416  054542  012601                  MOV     (SP)+,R1        ;;RESTORE R1
10417  054544  012600                  MOV     (SP)+,R0        ;;RESTORE R0
10418  054546  000207                  RTS     PC
10419  054550  000000          $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
10420                          ;;**************************************************************
10421
10422                          .SBTTL  POWER DOWN AND UP ROUTINES
10423
10424                          ;POWER DOWN ROUTINE
10425  054552  017737  124362  001404  $PWRDN: MOV     @SWR,SAVSWR     ;SAVE SWITCH REGISTER
10426  054560  012737  054572  000024          MOV     #$PWRUP,PWRVEC  ;SET UP VECTOR
10427  054566  000000                  HALT
10428  054570  000776                  BR      .-2             ;HANG UP
10429
10430                          ;POWER UP ROUTINE
10431  054572  005037  054676  $PWRUP: CLR     $PWRCT          ;WAIT LOOP FOR THE TTY
10432  054576  012737  000144  054700          MOV     #100.,$PWRCT+2
10433  054604  005237  054676  1$:     INC     $PWRCT          ;WAIT FOR THE INCREMENT
10434  054610  001375                  BNE     1$              ;   OF WORD
10435  054612  005337  054700          DEC     $PWRCT+2
10436  054616  001372                  BNE     1$
10437  054620  012737  054552  000024  MOV     #$PWRDN,PWRVEC  ;SET POWER DOWN VECTOR
10438  054626  012737  000340  000026  MOV     #PR7,PWRVEC+2   ;PRIORITY 7
10439  054634  012737  000340  000036  MOV     #PR7,TRAPVEC+2  ;LOCK OUT ALL INTERRUPTS FOR TRAPS
10440  054642  012706  001100          MOV     #STACK,SP       ;INITIALIZE STACK
10441  054646  104401  054702          TYPE    ,$POWER         ;REPORT POWER FAIL
10442  054652  004737  043670          JSR     PC,PRITIM       ;PRINT TIME
10443  054656  112737  177777  001361  MOVB    #-1,FLAG        ;SET FLAG FOR RESTART SEQUENCE
10444  054664  013777  001404  124246  MOV     SAVSWR,@SWR     ;RESTORE SWITCH REGISTER
10445  054672  000137  007172          JMP     PWRSRT          ;RESTART PROGRAM
10446
10447  054676  000000  000000  $PWRCT: .WORD   0,0             ;WAIT COUNT FOR TTY
10448  054702  005015  047520  042527  $POWER: .ASCIZ  <15><12>/POWER/<15><12>
10449  054710  006522  000012
10450                          .EVEN
10451
10452                          .SBTTL  INTEGER MULTIPLY ROUTINE
10453
10454                          ;;**************************************************************
```

```
10455                                   ;*
10456                                   ;*CALL
10457                                   ;*      MOV     MULTIPLER,-(SP)
10458                                   ;*      MOV     MULTIPLICAND,-(SP)
10459                                   ;*      JSR     PC,$MULT
10460                                   ;*      RETURN                          ;;PRODUCT IS ON THE STACK
10461                                   ;*
10462                                   ;*      STACK   PRODUCT
10463                                   ;*      -----   -------
10464                                   ;*
10465                                   ;*      TOP     LSB'S
10466                                   ;*      +2      MSB'S
10467                                   ;*
10468                                   ;;*****************************************************************
10469
10470  054714                           $MULT:
10471  054714  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10472  054716  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10473  054720  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10474  054722  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10475  054724  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10476  054726  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10477  054730  016605  000016                  MOV     16(SP),R5       ;STORE MULTIPLICAND
10478  054734  016603  000020                  MOV     20(SP),R3       ;STORE MULTIPLIER
10479  054740  005000                          CLR     R0              ;CLEAR HIGH ORDER WORDS
10480  054742  005001                          CLR     R1
10481  054744  005002                          CLR     R2
10482  054746  005004                          CLR     R4
10483  054750  012746  000041                  MOV     #41,-(SP)       ;MOVE 33 DEC TO COUNTER
10484  054754  006000                  1$:     ROR     R0
10485  054756  006001                          ROR     R1
10486  054760  006002                          ROR     R2              ;SHIFT TO ADD
10487  054762  006003                          ROR     R3
10488  054764  103003                          BCC     2$              ;NO CARRY NO ADD
10489  054766  060501                          ADD     R5,R1
10490  054770  005500                          ADC     R0              ;ADD DOUBLE PRECISION TO OBTAIN NEW PARTIAL
10491  054772  060400                          ADD     R4,R0           ;PRODUCT
10492  054774  005316                  2$:     DEC     (SP)            ;DECREMENT COUNTER
10493  054776  001366                          BNE     1$
10494  055000  005726                          TST     (SP)+           ;REMOVE COUNTER
10495  055002  010266  000020                  MOV     R2,20(SP)       ;GET LEAST SIGNIFICANT BITS
10496  055006  010366  000016                  MOV     R3,16(SP)       ;GET MOST SIGNIFICANT BITS
10497  055012  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
10498  055014  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
10499  055016  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
10500  055020  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10501  055022  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10502  055024  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10503  055026  000207                          RTS     PC              ;RETURN
10504
10505
10506                                   .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
10507
10508                                   ;;*****************************************************************
10509                                   ;*THIS ROUTINE IS A DOUBLE PRECISION PSEUDO RANDOM NUMBER GENERATOR
10510                                   ;*WITH A RANGE OF 0 TO 2(+33)-1.
```

```
10511                                   ;*CALL:
10512                                   ;*      JSR     PC,$RAND        ;;CALL THE ROUTINE
10513                                   ;*      RETURN                  ;;RETURN HERE THE RANDOM
10514                                   ;*                              ;;NUMBER WILL BE IN
10515                                   ;*                              ;;$HINUM,$LONUM
10516
10517  055030                           $RAND:
10518  055030  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10519  055032  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10520  055034  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10521  055036  013700  055130                  MOV     $LONUM,R0       ;;SET R0 WITH LOW
10522  055042  013701  055126                  MOV     $HINUM,R1       ;;SET R1 WITH HIGH
10523  055046  012702  177771                  MOV     #-7,R2          ;;SET SHIFT COUNT
10524  055052  006300                  1$:     ASL     R0              ;;SHIFT R0 LEFT AND
10525  055054  006101                          ROL     R1              ;;ROTATE CARRY INTO R1 AND
10526  055056  005202                          INC     R2              ;;CHECK FOR DONE
10527  055060  001374                          BNE     1$              ;;CONTINUE SHIFT LOOP
10528  055062  063700  055130                  ADD     $LONUM,R0       ;;ADD NUMBER TO MAKE X 129
10529  055066  005501                          ADC     R1              ;;PROPOGATE CARRY
10530  055070  063701  055126                  ADD     $HINUM,R1       ;;ADD NUMBER TO MAKE X 129
10531  055074  062700  001057                  ADD     #1057,R0        ;;ADD LOW CONSTANT
10532  055100  005501                          ADC     R1              ;;PROPOGATE CARRY
10533  055102  062701  047401                  ADD     #47401,R1       ;;ADD HIGH CONSTANT
10534  055106  010037  055130                  MOV     R0,$LONUM       ;;SAVE R0
10535  055112  010137  055126                  MOV     R1,$HINUM       ;;SAVE R1
10536  055116  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
10537  055120  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
10538  055122  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
10539  055124  000207                          RTS     PC              ;;RETURN
10540  055126  176543                  $HINUM: .WORD   176543
10541  055130  123456                  $LONUM: .WORD   123456
10542
10543                                   .SBTTL  SINGLE/DOUBLE LENGTH BINARY TO DECIMAL ASCII CONVERT ROUTINES
10544
10545                                   ;;*****************************************************************
10546                                   ;*
10547                                   ;*      THESE ROUTINE WIL CONVERT A 16-BIT OR 32-BIT BINARY NUMBER
10548                                   ;*      TO AN UNSIGNED DECIMAL (ASCII) NUMBER.
10549                                   ;*
10550                                   ;*CALL  MOV     #PNTR,-(SP)     ;POINTER TO LOW WORD OF BINARY NUMBER
10551                                   ;*      JSR     PC,$DB2D
10552                                   ;*      RETURN                  ;DECIMAL NUMBER IN $DECVL
10553                                   ;*
10554                                   ;*CALL  MOV     NUM,-(SP)
10555                                   ;*      JSR     PC,$SB2D
10556                                   ;*      RETURN                  ;DECMAL NUMBER IN $DECVL
10557                                   ;*
10558                                   ;;*****************************************************************
10559
10560  055132                           $SB2D:
10561  055132  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10562  055134  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10563  055136  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10564  055140  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10565  055142  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10566  055144  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
```

```
10567  055146  016601  000016                  MOV     16(SP),R1       ;STORE NUMBER IN R1
10568  055152  005002                          CLR     R2              ;CLEAR MOST SIGNIFICANT BITS
10569  055154  012737  000005  055300          MOV     #5,$DECNT       ;SET UP FOR 5 CONVERSIONS
10570  055162  012704  055356                  MOV     #$TNPW,R4       ;ADDRESS FOR 5 POWER
10571  055166  012705  055360                  MOV     #$TNPW+2,R5
10572  055172  000421                          BR      $B2D            ;START CONVERSION
10573
10574  055174                          $DB2D:
10575  055174  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10576  055176  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10577  055200  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
10578  055202  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
10579  055204  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
10580  055206  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
10581  055210  016602  000016                  MOV     16(SP),R2       ;PICKUP DATA POINTER
10582  055214  012201                          MOV     (R2)+,R1        ;PICKUP BINARY NUMBER
10583  055216  011202                          MOV     (R2),R2
10584  055220  012737  000012  055300          MOV     #10.,$DECNT     ;SET UP TO DO 10 CONVERSIONS
10585  055226  012704  055332                  MOV     #$TNPWR,R4      ;ADDRESS OF TEN POWER
10586  055232  012705  055334                  MOV     #$TNPWR+2,R5
10587
10588  055236  012700  055402          $B2D:   MOV     #$DECVL,R0      ;GET ADDRESS OF "$DECVL" STRING
10589  055242  005003          $2DEC:  CLR     R3              ;CLEAR PARTIAL
10590  055244  161401          2$:     SUB     (R4),R1         ;SUBTRACT TEN POWER
10591  055246  005602                  SBC     R2
10592  055250  161502                  SUB     (R5),R2
10593  055252  002402                  BLT     3$              ;BRANCH IF TEN POWER TOO LARGE
10594  055254  005203                  INC     R3              ;ADD 1 TO PARTIAL
10595  055256  000772                  BR      2$              ;LOOP
10596
10597  055260  062401          3$:     ADD     (R4)+,R1        ;RESTORE SUBTRACTED VALUE
10598  055262  005502                  ADC     R2
10599  055264  062402                  ADD     (R4)+,R2
10600  055266  022525                  CMP     (R5)+,(R5)+     ;MOVE TO NEXT TEN POWER
10601  055270  052703  000060          BIS     #'0,R3          ;CHANGE IT TO ASCII
10602  055274  110320                  MOVB    R3,(R0)+        ;SAVE IT
10603  055276  005327                  DEC     (PC)+           ;DONE?
10604  055300  000000          $DECNT: .WORD   0
10605  055302  001357                  BNE     $2DEC           ;BRANCH IF NO
10606  055304  105020                  CLRB    (R0)+           ;TERMINATOR
10607  055306  012605                  MOV     (SP)+,R5        ;;POP STACK INTO R5
10608  055310  012604                  MOV     (SP)+,R4        ;;POP STACK INTO R4
10609  055312  012603                  MOV     (SP)+,R3        ;;POP STACK INTO R3
10610  055314  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
10611  055316  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
10612  055320  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
10613  055322  012616                  MOV     (SP)+,(SP)      ;SETUP STACK FOR RETURN
10614  055324  104401  055402          TYPE    ,$DECVL
10615  055330  000207                  RTS     PC              ;RETURN
10616
10617  055332  145000          $TNPWR: 145000                  ;1.0E09
10618  055334  035632                  35632
10619  055336  160400                  160400                  ;1.0E08
10620  055340  002765                  2765
10621  055342  113200                  113200                  ;1.0E07
10622  055344  000230                  230
```

```
10623  055346  041100                          041100                  ;1.0E06
10624  055350  000017                          17
10625  055352  103240                          103240                  ;1.0E05
10626  055354  000001                          1
10627  055356  023420                  $TNPW:  23420                   ;1.0E04
10628  055360  000000                          0
10629  055362  001750                          1750                    ;1.0E03
10630  055364  000000                          0
10631  055366  000144                          144                     ;1.0E02
10632  055370  000000                          0
10633  055372  000012                          12                      ;1.0E01
10634  055374  000000                          0
10635  055376  000001                          1                       ;1.0E00
10636  055400  000000                          0
10637  055402  000014                  $DECVL: .BLKB   12.             ;RESERVE STORAGE FOR ASCIZ STRING
10638                                  .SBTTL  APT COMMUNICATIONS ROUTINE
10639
10640                                  ;;**************************************************************
10641  055416  112737  000001  055662  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
10642  055424  112737  000001  055660  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
10643  055432  000403                          BR      $ATYC
10644  055434  112737  000001  055662  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
10645  055442                          $ATYC:
10646  055442  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
10647  055444  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
10648  055446  105737  055660                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
10649  055452  001450                          BEQ     5$              ;;IF NOT:  BR
10650  055454  122737  000001  001210          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
10651  055462  001031                          BNE     3$              ;;IF NOT:  BR
10652  055464  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
10653  055472  001425                          BEQ     3$              ;;IF NOT:  BR
10654  055474  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
10655  055500  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
10656  055506  005737  001170          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
10657  055512  001375                          BNE     1$              ;;IF NOT:  WAIT
10658  055514  010037  001204                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
10659  055520  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
10660  055522  001376                          BNE     2$
10661  055524  163700  001204                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
10662  055530  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
10663  055532  010037  001206                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
10664  055536  012737  000004  001170          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
10665  055544  000413                          BR      5$
10666  055546  017637  000004  055572  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
10667  055554  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
10668  055562  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
10669  055566  004737  052512                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
10670  055572  000000                  4$:     .WORD   0
10671  055574                          5$:
10672  055574  105737  055662          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
10673  055600  001416                          BEQ     12$             ;;IF NOT:  BR
10674  055602  005737  001210                  TST     $ENV            ;;RUNNING UNDER APT?
10675  055606  001413                          BEQ     12$             ;;IF NOT:  BR
10676  055610  005737  001170          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
10677  055614  001375                          BNE     11$             ;;IF NOT:  WAIT
10678  055616  017637  000004  001172          MOV     @4(SP),$FATAL   ;;GET ERROR #
```

```
10679  055624  062766  000002  000004          ADD     #2,4(SP)            ;;BUMP RETURN ADDR.
10680  055632  005237  001170                  INC     $MSGTYPE            ;;TELL APT TO TAKE ERROR
10681  055636  105037  055662          12$:    CLRB    $FFLG               ;;CLEAR FATAL FLAG
10682  055642  105037  055661                  CLRB    $LFLG               ;;CLEAR LOG FLAG
10683  055646  105037  055660                  CLRB    $MFLG               ;;CLEAR MESSAGE FLAG
10684  055652  012601                          MOV     (SP)+,R1            ;;POP STACK INTO R1
10685  055654  012600                          MOV     (SP)+,R0            ;;POP STACK INTO R0
10686  055656  000207                          RTS     PC                  ;;RETURN
10687  055660     000          $MFLG:  .BYTE   0                           ;;MESSG. FLAG
10688  055661     000          $LFLG:  .BYTE   0                           ;;LOG FLAG
10689  055662     000          $FFLG:  .BYTE   0                           ;;FATAL FLAG
10690          055664                  .EVEN
10691          000200          APTSIZE=200
10692          000001          APTENV=001
10693          000100          APTSPOOL=100
10694          000040          APTCSUP=040
10695                          .SBTTL  TRAP DECODER
10696
10697                          ;;**************************************************************
10698                          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
10699                          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
10700                          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
10701                          ;*GO TO THAT ROUTINE.
10702
10703  055664  010046          $TRAP:  MOV     R0,-(SP)            ;;SAVE R0
10704  055666  016600  000002          MOV     2(SP),R0            ;;GET TRAP ADDRESS
10705  055672  005740                  TST     -(R0)               ;;BACKUP BY 2
10706  055674  111000                  MOVB    (R0),R0             ;;GET RIGHT BYTE OF TRAP
10707  055676  006300                  ASL     R0                  ;;POSITION FOR INDEXING
10708  055700  016000  055720          MOV     $TRPAD(R0),R0       ;;INDEX TO TABLE
10709  055704  000200                  RTS     R0                  ;;GO TO ROUTINE
10710
10711
10712                          ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
10713
10714  055706  011646          $TRAP2: MOV     (SP),-(SP)          ;;MOVE THE PC DOWN
10715  055710  016666  000004  000002  MOV     4(SP),2(SP)         ;;MOVE THE PSW DOWN
10716  055716  000002                  RTI                         ;;RESTORE THE PSW
10717
10718                          .SBTTL  TRAP TABLE
10719
10720                          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
10721                          ;*BY THE "TRAP" INSTRUCTION.
10722
10723                          ;       ROUTINE
10724                          ;       -------
10725  055720  055706          $TRPAD: .WORD   $TRAP2
10726  055722  052512                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
10727
10728
10729  055724  053100                  $RDCHR  ;;CALL=RDCHR    TRAP+2(104402)  TTY TYPEIN CHARACTER ROUTINE
10730  055726  053220                  $RDLIN  ;;CALL=RDLIN    TRAP+3(104403)  TTY TYPEIN STRING ROUTINE
```

```
10731                                           .SBTTL  ASCII MESSAGES
10732
10733  055730  005015  044103  047101  XXDPMG: .ASCII  <CR><LF>/CHANGE XXDP PACK/
10734  055736  042507  054040  042130
10735  055744  020120  040520  045503
10736  055752  005015  046103  040505          .ASCIZ  <CR><LF>/CLEAR LOC 40,RESTART PROGRAM/
10737  055760  020122  047514  020103
10738  055766  030064  051054  051505
10739  055774  040524  052122  050040
10740  056002  047522  051107  046501
10741  056010     000
10742  056011     111  046114  043505  ILLCOM: .ASCII  /ILLEGAL OPERATOR COMMAND /
10743  056016  046101  047440  042520
10744  056024  040522  047524  020122
10745  056032  047503  046515  047101
10746  056040  020104
10747  056042  020040  006477  000012  ILLCMD: .ASCIZ  /  ?/<15><12>
10748  056050  051104  053111  020105  OPR000: .ASCII  /DRIVE NUMBER /
10749  056056  052516  041115  051105
10750  056064     040
10751  056065     060  000040          OPR001: .ASCIZ  /0 /
10752  056070  046111  042514  040507  OPR002: .ASCIZ  /ILLEGAL?/<15><12>
10753  056076  037514  005015     000
10754  056103     125  042116  051105  OPR003: .ASCIZ  /UNDER TEST/
10755  056110  052040  051505  000124
10756  056116  046101  042522  042101  OPR004: .ASCIZ  /ALREADY ASSIGNED?/<15><12>
10757  056124  020131  051501  044523
10758  056132  047107  042105  006477
10759  056140  000012
10760  056142  047516  020124  051501  OPR005: .ASCIZ  /NOT ASSIGNED?/<15><12>
10761  056150  044523  047107  042105
10762  056156  006477  000012
10763  056162  040510  020123  042502  OPR006: .ASCIZ  /HAS BEEN DROPPED FROM TEST SEQUENCE/<15><12>
10764  056170  047105  042040  047522
10765  056176  050120  042105  043040
10766  056204  047522  020115  042524
10767  056212  052123  051440  050505
10768  056220  042525  041516  006505
10769  056226  000012
10770  056230  046120  040505  042523  OPR007: .ASCIZ  /PLEASE ENTER COMMAND/<15><12>/*/
10771  056236  042440  052116  051105
10772  056244  041440  046517  040515
10773  056252  042116  005015  000052
10774  056260  050117  051105  052101  OPR008: .ASCIZ  /OPERATOR INITIATED DROP DRIVE DURING WRITE PACK/
10775  056266  051117  044440  044516
10776  056274  044524  052101  042105
10777  056302  042040  047522  020120
10778  056310  051104  053111  020105
10779  056316  052504  044522  043516
10780  056324  053440  044522  042524
10781  056332  050040  041501  000113
10782  056340  005015  051104  053111  OPR010: .ASCII  <15><12>/DRIVE NUMBER /
10783  056346  020105  052516  041115
10784  056354  051105     040
10785  056357     060  000040          OPR011: .ASCIZ  /0 /
10786  056362  051104  053111  020105  OPR012: .ASCIZ  /DRIVE WRITE LOCKED/<15><12>
```

```
10787  056370  051127  052111  020105
10788  056376  047514  045503  042105
10789  056404  005015     000
10790  056407     101  046114  042040  OPR013: .ASCIZ  /ALL DRIVES CURRENTLY UNDER TEST/<15><12>
10791  056414  044522  042526  020123
10792  056422  052503  051122  047105
10793  056430  046124  020131  047125
10794  056436  042504  020122  042524
10795  056444  052123  005015     000
10796  056451     116  020117  051104  OPR014: .ASCIZ  /NO DRIVE IN USE/<15><12>
10797  056456  053111  020105  047111
10798  056464  052440  042523  005015
10799  056472     000
10800  056473     101  046114  043511  OPR015: .ASCIZ  /ALLIGNMENT PACK IN DRIVE/<15><12>
10801  056500  046516  047105  020124
10802  056506  040520  045503  044440
10803  056514  020116  051104  053111
10804  056522  006505  000012
10805  056526  025052  051104  053111  DRVSTT: .ASCIZ  /**DRIVE STATISTICS**/<15><12>
10806  056534  020105  052123  052101
10807  056542  051511  044524  051503
10808  056550  025052  005015     000
10809  056555     015  042012  044522  STT000: .ASCII  <15><12>/DRIVE SERIAL NO. /
10810  056562  042526  051440  051105
10811  056570  040511  020114  047516
10812  056576  020056
10813  056600  000003                  STT100: .BLKB   3
10814  056603     000                          .BYTE   0
10815  056604  051117  042504  051522  STT001: .ASCIZ  /ORDERS PERFORMED /
10816  056612  050040  051105  047506
10817  056620  046522  042105  000040
10818  056626  005015  047527  042122  STT002: .ASCIZ  <15><12>/WORDS WRITTEN*65K /
10819  056634  020123  051127  052111
10820  056642  042524  025116  032466
10821  056650  020113     000
10822  056653     015  053412  051117  STT003: .ASCIZ  <15><12>/WORDS READ*65K /
10823  056660  051504  051040  040505
10824  056666  025104  032466  020113
10825  056674     000
10826  056675     015  051412  043117  STT004: .ASCIZ  <15><12>/SOFT DATA ERRORS/<15><12>/  ECC /
10827  056702  020124  040504  040524
10828  056710  042440  051122  051117
10829  056716  006523  020012  042440
10830  056724  041503  000040
10831  056730  005015  020040  042522  STT005: .ASCIZ  <15><12>/  REREAD /
10832  056736  042522  042101  000040
10833  056744  005015  020040  043117  STT006: .ASCIZ  <15><12>/  OFFSET /
10834  056752  051506  052105  000040
10835  056760  005015  040510  042122  STT007: .ASCIZ  <15><12>/HARD DATA ERRORS /
10836  056766  042040  052101  020101
10837  056774  051105  047522  051522
10838  057002  000040
10839  057004  005015  042523  045505  STT008: .ASCIZ  <15><12>/SEEK INCOMPLETES /
10840  057012  044440  041516  046517
10841  057020  046120  052105  051505
10842  057026  000040
```

```
10843  057030  005015  050117  051105  STT009: .ASCIZ  <15><12>"OPERATION INCOMPLETES/MISPOSITIONS "
10844  057036  052101  047511  020116
10845  057044  047111  047503  050115
10846  057052  042514  042524  027523
10847  057060  044515  050123  051517
10848  057066  052111  047511  051516
10849  057074  000040
10850  057076  005015  052117  042510  STT010: .ASCIZ  <15><12>/OTHER ERRORS /
10851  057104  020122  051105  047522
10852  057112  051522  000040
10853  057116  005015  040520  045503  STT011: .ASCII  <15><12>/PACK SERIAL NUMBER /
10854  057124  051440  051105  040511
10855  057132  020114  052516  041115
10856  057140  051105     040
10857  057143  000013                  STT012: .BLKB   11.
10858  057156  005015     000          STT013: .ASCIZ  <15><12>
10859  057161     103  052520  044440  SYS000: .ASCIZ  /CPU ID=/
10860  057166  036504     000
10861  057171     117  042526  046122  SYS001: .ASCIZ  /OVERLAY LOADER=/
10862  057176  054501  046040  040517
10863  057204  042504  036522     000
10864  057211     115  054101  052040  SYS002: .ASCIZ  /MAX TRANSFER SIZE ( MAX: /
10865  057216  040522  051516  042506
10866  057224  020122  044523  042532
10867  057232  024040  046440  054101
10868  057240  020072     000
10869  057243     040  036451     000  SYS003: .ASCIZ  / )=/
10870  057247     116  027117  047440  SYS004: .ASCIZ  /NO. OF SOFTWARE COMPARES=/
10871  057254  020106  047523  052106
10872  057262  040527  042522  041440
10873  057270  046517  040520  042522
10874  057276  036523     000
10875  057301     122  030113  026466  SYS005: .ASCIZ  /RK06-RK07 BUS ADD=/
10876  057306  045522  033460  041040
10877  057314  051525  040440  042104
10878  057322  000075
10879  057324  045522  033060  051055  SYS006: .ASCIZ  /RK06-RK07 VEC ADD=/
10880  057332  030113  020067  042526
10881  057340  020103  042101  036504
10882  057346     000
10883  057347     116  047117  054105  SYS007: .ASCIZ  /NONEXISTENT MEMORY/<15><12>
10884  057354  051511  042524  052116
10885  057362  046440  046505  051117
10886  057370  006531  000012
10887  057374  042522  042101  020131  SYS008: .ASCIZ  /READY TO BEGIN PERFORMANCE TESTING/<15><12><12>
10888  057402  047524  041040  043505
10889  057410  047111  050040  051105
10890  057416  047506  046522  047101
10891  057424  042503  052040  051505
10892  057432  044524  043516  005015
10893  057440  000012
10894  057442  052123  052101  051511  SYS009: .ASCIZ  /STATISTIC INTERVAL=/
10895  057450  044524  020103  047111
10896  057456  042524  053122  046101
10897  057464  000075
10898  057466  045522  033060  051055  SYS010: .ASCIZ  /RK06-RK07 PRIOR=/
```

```
10899  057474  030113  020067  051120
10900  057502  047511  036522     000
10901  057507     120  051101  042515  PAR000: .ASCII  /PARMETERS FOR DRIVE /
10902  057514  042524  051522  043040
10903  057522  051117  042040  044522
10904  057530  042526     040
10905  057533     060  005015  000012  PAR001: .ASCIZ  /0/<15><12><12>
10906  057540  054503  044514  042116  PAR002: .ASCIZ  /CYLINDER MAX,MIN =/
10907  057546  051105  046440  054101
10908  057554  046454  047111  036440
10909  057562     000
10910  057563     124  040522  045503  PAR003: .ASCIZ  /TRACK MAX,MIN =/
10911  057570  046440  054101  046454
10912  057576  047111  036440     000
10913  057603     123  041505  047524  PAR004: .ASCIZ  /SECTOR MAX,MIN =/
10914  057610  020122  040515  026130
10915  057616  044515  020116  000075
10916  057624  042522  042101  053457  PAR005: .ASCIZ  "READ/WRITE RATIO ="
10917  057632  044522  042524  051040
10918  057640  052101  047511  036440
10919  057646     000
10920  057647     127  044522  042524  PAR006: .ASCIZ  /WRITE CHECK AFTER WRITE =/
10921  057654  041440  042510  045503
10922  057662  040440  052106  051105
10923  057670  053440  044522  042524
10924  057676  036440     000
10925  057701     103  051117  042522  PAR007: .ASCIZ  /CORRECTABLE READ ERROR THRESHOLD =/
10926  057706  052103  041101  042514
10927  057714  051040  040505  020104
10928  057722  051105  047522  020122
10929  057730  044124  042522  044123
10930  057736  046117  020104  000075
10931  057744  047125  047503  051122  PAR008: .ASCIZ  /UNCORRECTABLE READ ERROR THRESHOLD =/
10932  057752  041505  040524  046102
10933  057760  020105  042522  042101
10934  057766  042440  051122  051117
10935  057774  052040  051110  051505
10936  060002  047510  042114  036440
10937  060010     000
10938  060011     123  042505  020113  PAR009: .ASCIZ  /SEEK ERROR THRESHOLD =/
10939  060016  051105  047522  020122
10940  060024  044124  042522  044123
10941  060032  046117  020104  000075
10942  060040  047111  044510  044502  PAR010: .ASCIZ  /INHIBIT ERROR CORRECTION AND RETRY =/
10943  060046  020124  051105  047522
10944  060054  020122  047503  051122
10945  060062  041505  044524  047117
10946  060070  040440  042116  051040
10947  060076  052105  054522  036440
10948  060104     000
10949  060105     117  042520  040522  PAR011: .ASCIZ  /OPERATION COUNT THRESHOLD*65K =/
10950  060112  044524  047117  041440
10951  060120  052517  052116  052040
10952  060126  051110  051505  047510
10953  060134  042114  033052  045465
10954  060142  036440     000
```

```
10955  060145     127  051117  051504  PAR012: .ASCIZ  /WORDS TRANSFERRED THRESHOLD*1,049K =/
10956  060152  052040  040522  051516
10957  060160  042506  051122  042105
10958  060166  052040  051110  051505
10959  060174  047510  042114  030452
10960  060202  030054  034464  020113
10961  060210  000075
10962  060212  044506  051522  000124  PAR013: .ASCIZ  /FIRST/
10963  060220  042523  047503  042116  PAR014: .ASCIZ  /SECOND/
10964  060226     000
10965  060227     124  044510  042122  PAR015: .ASCIZ  /THIRD/
10966  060234     000
10967  060235     106  052517  052122  PAR016: .ASCIZ  /FOURTH/
10968  060242  000110
10969  060244  044506  052106  000110  PAR017: .ASCIZ  /FIFTH/
10970  060252  042440  041530  052514  PAR018: .ASCIZ  / EXCLUDED PACK AREA =/
10971  060260  042504  020104  040520
10972  060266  045503  040440  042522
10973  060274  020101  000075
10974  060300  044103  047101  042507  PAR019: .ASCIZ  /CHANGE/
10975  060306     000
10976  060307     125  040516  046102  PAR020: .ASCII  /UNABLE TO GENERATE NEW COMMAND DUE TO/<15><12>
10977  060314  020105  047524  043440
10978  060322  047105  051105  052101
10979  060330  020105  042516  020127
10980  060336  047503  046515  047101
10981  060344  020104  052504  020105
10982  060352  047524  005015
10983  060356  054105  046103  042125          .ASCIZ  /EXCLUDED PACK AREAS AND MAX TRANSFER SIZE CONFLICT/<15><12>
10984  060364  042105  050040  041501
10985  060372  020113  051101  040505
10986  060400  020123  047101  020104
10987  060406  040515  020130  051124
10988  060414  047101  043123  051105
10989  060422  051440  055111  020105
10990  060430  047503  043116  044514
10991  060436  052103  005015     000
10992  060443     123  046501  046120  PAR021: .ASCIZ  /SAMPLED COMPARES =/
10993  060450  042105  041440  046517
10994  060456  040520  042522  020123
10995  060464  000075
10996  060466  042524  052123  047111  PAR022: .ASCIZ  /TESTING RK06? Y OR N  /
10997  060474  020107  045522  033060
10998  060502  020077  020131  051117
10999  060510  047040  020040     000
11000  060515     040  037440  037477  QUESMK: .ASCIZ  /  ??????  /
11001  060522  037477  020077  000040
11002  060530  020040     000          BLANKS: .ASCIZ  /  /
11003  060533     052  025052  040506  ERR000: .ASCIZ  /***FATAL ERROR***/<15><12><12>
11004  060540  040524  020114  051105
11005  060546  047522  025122  025052
11006  060554  005015  000012
11007  060560  047503  052116  047522  ERR001: .ASCIZ  /CONTROLLER CLEAR DID NOT CLEAR ERROR/
11008  060566  046114  051105  041440
11009  060574  042514  051101  042040
11010  060602  042111  047040  052117
```

```
11011  060610  041440  042514  051101
11012  060616  042440  051122  051117
11013  060624     000
11014  060625     116  020117  052101  ERR002: .ASCIZ  /NO ATTENTION IN RKASOF/
11015  060632  042524  052116  047511
11016  060640  020116  047111  051040
11017  060646  040513  047523  000106
11018  060654  047125  054105  042520  ERR003: .ASCIZ  /UNEXPECTED DATA TYPE ERROR/
11019  060662  052103  042105  042040
11020  060670  052101  020101  054524
11021  060676  042520  042440  051122
11022  060704  051117     000
11023  060707     104  044522  042526  ERR004: .ASCIZ  /DRIVE CLEAR DID NOT RESET ATTENTION/
11024  060714  041440  042514  051101
11025  060722  042040  042111  047040
11026  060730  052117  051040  051505
11027  060736  052105  040440  052124
11028  060744  047105  044524  047117
11029  060752     000
11030  060753     111  046114  043505  ERR005: .ASCIZ  /ILLEGAL DRIVER COMMAND/
11031  060760  046101  042040  044522
11032  060766  042526  020122  047503
11033  060774  046515  047101  000104
11034  061002  052515  052114  050111  ERR006: .ASCIZ  /MULTIPLE DRIVE SELECT/
11035  061010  042514  042040  044522
11036  061016  042526  051440  046105
11037  061024  041505  000124
11038  061030  042523  020124  053523  ERR007: .ASCIZ  /SET SWITCH 14 TO CYCLE ON ERROR/<15><12>
11039  061036  052111  044103  030440
11040  061044  020064  047524  041440
11041  061052  041531  042514  047440
11042  061060  020116  051105  047522
11043  061066  006522  000012
11044  061072  047503  052116  030122  ERR008: .ASCIZ  /CONTROLLER ERROR THRESHOLD EXCEEDED/
11045  061100  046114  051105  042440
11046  061106  051122  051117  052040
11047  061114  051110  051505  047510
11048  061122  042114  042440  041530
11049  061130  042505  042504  000104
11050  061136  040504  040524  046040  ERR009: .ASCIZ  /DATA LATE THRESHOLD EXCEEDED/
11051  061144  052101  020105  044124
11052  061152  042522  044123  046117
11053  061160  020104  054105  042503
11054  061166  042105  042105     000
11055  061173     101  042104  042522  ERR012: .ASCII  /ADDRESS GOOD    BAD     SECTOR  DATA/<15><12>
11056  061200  051523  043440  047517
11057  061206  020104  020040  041040
11058  061214  042101  020040  020040
11059  061222  051440  041505  047524
11060  061230  020122  042040  052101
11061  061236  006501     012
11062  061241     040  020040  020040          .ASCIZ  /        DATA    DATA    POS     PAT #/<15><12>
11063  061246  020040  042040  052101
11064  061254  020101  020040  042040
11065  061262  052101  020101  020040
11066  061270  050040  051517  020040
```

```
11067  061276  020040  050040  052101
11068  061304  021440  005015     000
11069  061311     103  047117  051124  ERR013: .ASCIZ  /CONTROLLER ERROR NOT FLAGGED/<15><12>
11070  061316  046117  042514  020122
11071  061324  051105  047522  020122
11072  061332  047516  020124  046106
11073  061340  043501  042507  006504
11074  061346  000012
11075  061350  047527  042122  041440  ERR014: .ASCIZ  /WORD COUNT NOT EQUAL 0/<15><12>
11076  061356  052517  052116  047040
11077  061364  052117  042440  052521
11078  061372  046101  030040  005015
11079  061400     000
11080  061401     102  051525  040440  ERR015: .ASCIZ  /BUS ADD INCORRECT/<15><12>
11081  061406  042104  044440  041516
11082  061414  051117  042522  052103
11083  061422  005015     000
11084  061425     103  046131  052054  ERR016: .ASCIZ  /CYL,TRK,SEC INCORRECT/<15><12>
11085  061432  045522  051454  041505
11086  061440  044440  041516  051117
11087  061446  042522  052103  005015
11088  061454     000
11089  061455     103  047117  051124  ERR017: .ASCIZ  /CONTROLLER COMMAND TIME OUT/
11090  061462  046117  042514  020122
11091  061470  047503  046515  047101
11092  061476  020104  044524  042515
11093  061504  047440  052125     000
11094  061511     120  047522  020107  ERR020: .ASCIZ  /PROG ERROR/
11095  061516  051105  047522  000122  ERR021: .ASCIZ  /ILL FUNCT CODE/
11096  061524  046111  020114  052506
11097  061532  041516  020124  047503
11098  061540  042504     000
11099  061543     106                  ERR022: .BYTE   106
11100  061544  051105  047522  020122          .ASCIZ  /ERROR WHILE WAITING TO REPORT ERROR/
11101  061552  044127  046111  020105
11102  061560  040527  052111  047111
11103  061566  020107  047524  051040
11104  061574  050105  051117  020124
11105  061602  051105  047522  000122
11106  061610     106                  ERR023: .BYTE   106
11107  061611     116  047117  042455          .ASCIZ  /NON-EXIST DRV/
11108  061616  044530  052123  042040
11109  061624  053122     000
11110  061627     106                  ERR024: .BYTE   106
11111  061630  042523  041522  047117          .ASCIZ  /SERCON PAR/
11112  061636  050040  051101     000
11113  061643     106                  ERR025: .BYTE   106
11114  061644  047125  052111  043040          .ASCIZ  /UNIT FIELD ERR/
11115  061652  042511  042114  042440
11116  061660  051122     000
11117  061663     015  044412  052116  ERR026: .ASCIZ  <15><12>/INT FROM DRV NOT UNDER TEST/<15><12>
11118  061670  043040  047522  020115
11119  061676  051104  020126  047516
11120  061704  020124  047125  042504
11121  061712  020122  042524  052123
11122  061720  005015     000
```

```
11123  061723  110     051101  020104  ERR027: .ASCIZ  /HARD DRV ERR/<15><12>
11124  061730  051104  020126  051105
11125  061736  006522  000012
11126  061742  051104  020126  052123  ERR028: .ASCIZ  /DRV STATUS CHANGE DID NOT CLR/<15><12>
11127  061750  052101  051525  041440
11128  061756  040510  043516  020103
11129  061764  044504  020104  047516
11130  061772  020124  046103  006522
11131  062000  000012
11132  062002  106                   ERR029: .BYTE   106
11133  062003  104     044522  042526         .ASCIZ  /DRIVE BECAME NOT AVAIL/
11134  062010  041040  041505  046501
11135  062016  020105  047516  020124
11136  062024  053101  044501  000114
11137  062032  047125  054105  042520  ERR030: .ASCIZ  /UNEXPECT ATTN/<15><12>
11138  062040  052103  040440  052124
11139  062046  006516  000012
11140  062052  046111  020114  051504  ERR031: .ASCIZ  /ILL DSK ADD/
11141  062060  020113  042101  000104
11142  062066  051127  020124  047514  ERR032: .ASCIZ  /WRT LOCK ERR/
11143  062074  045503  042440  051122
11144  062102  000
11145  062103  101     020103  047514  ERR033: .ASCIZ  /AC LOW/
11146  062110  000127
11147  062112  050123  042505  020104  ERR034: .ASCIZ  /SPEED LOSS/
11148  062120  047514  051523  000
11149  062125  104     053122  052040  ERR035: .ASCIZ  /DRV TYPE ERR/
11150  062132  050131  020105  051105
11151  062140  000122
11152  062142  047506  046522  052101  ERR036: .ASCIZ  /FORMAT ERR/
11153  062150  042440  051122  000
11154  062155  116     047117  042455  ERR037: .ASCIZ  /NON-EXIST DRV FUNCT/
11155  062162  044530  052123  042040
11156  062170  053122  043040  047125
11157  062176  052103  000
11158  062201  106                   ERR038: .BYTE   106
11159  062202  042523  045505  044440         .ASCIZ  /SEEK INCOMP/
11160  062210  041516  046517  000120
11161  062216  106                   ERR039: .BYTE   106
11162  062217  104     053122  047440         .ASCIZ  /DRV OFF TRK/
11163  062224  043106  052040  045522
11164  062232  000
11165  062233  106                   ERR040: .BYTE   106
11166  062234  044515  050123  051517         .ASCIZ  /MISPOS/
11167  062242  000
11168  062243  126                   ERR042: .BYTE   126
11169  062244  050117  051105  052101         .ASCIZ  /OPERAT INCOMP/
11170  062252  044440  041516  046517
11171  062260  000120
11172  062262  127                   ERR043: .BYTE   127
11173  062263  110     051104  053040         .ASCIZ  /HDR VRC ERR/
11174  062270  041522  042440  051122
11175  062276  000
11176  062277  116     047117  042455  ERR044: .ASCIZ  /NON-EXIST MEM/
11177  062304  044530  052123  046440
11178  062312  046505  000
```

```
11179  062315  000125  044516  052502  ERR045: .ASCIZ  /UNIBUS PAR/
11180  062322  020123  040520  000122
11181  062330  000117                  ERR046: .BYTE   117
11182  062331  000104  053122  052040          .ASCIZ  /DRV TIMING ERR/
11183  062336  046511  047111  020107
11184  062344  051105  000122
11185  062350  040504  040524  046040  ERR047: .ASCIZ  /DATA LATE/<15><12>
11186  062356  052101  006505  000012
11187  062364  000117                  ERR048: .BYTE   117
11188  062365  000104  052101  020101          .ASCIZ  /DATA CHK/
11189  062372  044103  000113
11190  062376  000107                  ERR050: .BYTE   107
11191  062377  000127  044522  042524          .ASCIZ  /WRITE CHK ERR/
11192  062404  041440  045510  042440
11193  062412  051122  000000
11194  062415  000103  046131  040440  ERR051: .ASCIZ  /CYL ADDR OVRFLW/
11195  062422  042104  020122  053117
11196  062430  043122  053514  000000
11197  062435  000106                  ERR052: .BYTE   106
11198  062436  046503  042116  052040          .ASCIZ  /CMND TIME OUT DURING POSITIONING OPERATION/<15><12>
11199  062444  046511  020105  052517
11200  062452  020124  052504  044522
11201  062460  043516  050040  051517
11202  062466  052111  047511  044516
11203  062474  043516  047440  042520
11204  062502  040522  044524  047117
11205  062510  005015  000000
11206  062513  000106                  ERR053: .BYTE   106
11207  062514  051104  020126  047516          .ASCIZ  /DRV NOT RDY AFTER START SPINDLE/
11208  062522  020124  042122  020131
11209  062530  043101  042524  020122
11210  062536  052123  051101  020124
11211  062544  050123  047111  046104
11212  062552  000105
11213  062554  000106                  ERR054: .BYTE   106
11214  062555  000126  046117  053040          .ASCIZ  /VOL VALID DID NOT SET AFTER PACK ACK/
11215  062562  046101  042111  042040
11216  062570  042111  047040  052117
11217  062576  051440  052105  040440
11218  062604  052106  051105  050040
11219  062612  041501  020113  041501
11220  062620  000113
11221  062622  051104  020126  047125  ERR056: .ASCIZ  /DRV UNSAFE/
11222  062630  040523  042506  000000
11223  062635  000104  053122  042440  ERR057: .ASCIZ  /DRV ERR NOT INDICATED BY FAULT/<15><12>
11224  062642  051122  047040  052117
11225  062650  044440  042116  041511
11226  062656  052101  042105  041040
11227  062664  020131  040506  046125
11228  062672  006524  000012
11229  062676  051104  020126  052101  ERR058: .ASCIZ  /DRV ATTN BUT NO FAULT OR DRV STATUS CHANGE/<15><12>
11230  062704  047124  041040  052125
11231  062712  047040  020117  040506
11232  062720  046125  020124  051117
11233  062726  042040  053122  051440
11234  062734  040524  052524  020123
```

```
11235  062742  044103  047101  042507
11236  062750  005015     000
11237  062753     103  047115  020104  ERR059: .ASCIZ  /CMND TIME OUT DRV SEIZED BY OTHER PORT/<15><12>
11238  062760  044524  042515  047440
11239  062766  052125  042040  053122
11240  062774  051440  044505  042532
11241  063002  020104  054502  047440
11242  063010  044124  051105  050040
11243  063016  051117  006524  000012
11244  063024  041505  020103  044515  ERR060: .ASCIZ  /ECC MISCORRECTION/<15><12>
11245  063032  041523  051117  042522
11246  063040  052103  047511  006516
11247  063046  000012
11248  063050  040504  040524  041440  ERR062: .ASCIZ  /DATA COMPARE ERROR/<15><12>
11249  063056  046517  040520  042522
11250  063064  042440  051122  051117
11251  063072  005015     000
11252  063075     126          ERR064: .BYTE   126
11253  063076  044515  050123  051517          .ASCIZ  /MISPOS/
11254  063104     000
11255  063105     106          ERR065: .BYTE   106
11256  063106  042510  042101  051440          .ASCIZ  /HEAD SELECT ERROR/
11257  063114  046105  041505  020124
11259  063122  051105  047522  000122
11259  063130  041505  020103  047514  ERR066: .ASCIZ  /ECC LOGIC ERROR/
11260  063136  044507  020103  051105
11261  063144  047522  000122
11262  063150  045522  041505  052120  ERR067: .ASCIZ  /RKECPT  RKECPS/<15><12>
11263  063156  020040  045522  041505
11264  063164  051520  005015     000
11265  063171     127  051117  020104  ERR070: .ASCIZ  /WORD COUNT INVALID/
11266  063176  047503  047125  020124
11267  063204  047111  040526  044514
11268  063212  000104
11269  063214  052502  020123  042101  ERR071: .ASCIZ  /BUS ADDRESS INVALID/
11270  063222  051104  051505  020123
11271  063230  047111  040526  044514
11272  063236  000104
11273  063240  054503  044514  042116  ERR072: .ASCIZ  /CYLINDER INVALID/
11274  063246  051105  044440  053116
11275  063254  046101  042111     000
11276  063261     124  040522  045503  ERR073: .ASCIZ  "TRACK/SECTOR INVALID"
11277  063266  051457  041505  047524
11278  063274  020122  047111  040526
11279  063302  044514  000104
11280  063306     106          ERR074: .BYTE   106
11281  063307     105  051122  051117          .ASCIZ  /ERROR IN DRV DID NOT RELOAD HEADS/
11282  063314  044440  020116  051104
11283  063322  020126  044504  020104
11284  063330  047516  020124  042522
11285  063336  047514  042101  044040
11286  063344  040505  051504     000
11287  063351     103  051125  042522  ERR100: .ASCII  /CURRENT SUPPLIED/<15><12>
11288  063356  052116  051440  050125
11289  063364  046120  042511  006504
11290  063372     012
```

```
11291  063373     103  047115  020104          .ASCIZ  /CMND    CYL     TRK     SEC     OFFSET  BUS AD  WRD CT/
11292  063400  020040  041440  046131
11293  063406  020040  020040  052040
11294  063414  045522  020040  020040
11295  063422  051440  041505  020040
11296  063430  020040  047440  043106
11297  063436  042523  020124  041040
11298  063444  051525  040440  020104
11299  063452  053440  042122  041440
11300  063460  000124
11301  063462  051120  053105  047511  ERR101: .ASCII  /PREVIOUS /<15><12>
11302  063470  051525  006440      012
11303  063475     103  047115  020104          .ASCIZ  /CMND    CYL     TRK     SEC     WRD CT/
11304  063502  020040  041440  046131
11305  063510  020040  020040  052040
11306  063516  045522  020040  020040
11307  063524  051440  041505  020040
11308  063532  020040  053440  042122
11309  063540  041440  000124
11310  063544  020040  040520  000124  ERR102: .ASCIZ  /  PAT/
11311  063552  043117  051506  052105  ERR108: .ASCIZ  /OFFSET =/
11312  063560  036440     000
11313  063563     110  030504  020040  ERR111: .ASCIZ  /HD1     HD2     HD3/<15><12>
11314  063570  020040  044040  031104
11315  063576  020040  020040  044040
11316  063604  031504  005015     000
11317  063611     104  052101  020101  ERR200: .ASCIZ  /DATA LATE WHEN UNLOADING HEADER/
11318  063616  040514  042524  053440
11319  063624  042510  020116  047125
11320  063632  047514  042101  047111
11321  063640  020107  042510  042101
11322  063646  051105     000
11323  063651     103  047117  051124  ERR201: .ASCIZ  /CONTROLLER ERROR DURING DRIVE SERVICING/
11324  063656  046117  042514  020122
11325  063664  051105  047522  020122
11326  063672  052504  044522  043516
11327  063700  042040  044522  042526
11328  063706  051440  051105  044526
11329  063714  044503  043516     000
11330  063721     105  051122  051117  ERR202: .ASCIZ  /ERROR DURING ERROR RECOVERY/<15><12>
11331  063726  042040  051125  047111
11332  063734  020107  051105  047522
11333  063742  020122  042522  047503
11334  063750  042526  054522  005015
11335  063756     000
11336  063757     105  051122  051117  ERR203: .ASCIZ  /ERROR RECOVERY SUCESSFUL/<15><12>
11337  063764  051040  041505  053117
11338  063772  051105  020131  052523
11339  064000  042503  051523  052506
11340  064006  006514  000012
11341  064012  051105  047522  020122  ERR204: .ASCIZ  /ERROR RECOVERY UNSUCESSFUL/<15><12>
11342  064020  042522  047503  042526
11343  064026  054522  052440  051516
11344  064034  041525  051505  043123
11345  064042  046125  005015     000
11346  064047     105  051122  051117  ERR205: .ASCIZ  /ERROR THRESHOLD EXCEEDED/<15><12>
```

```
11347  064054  052040  051110  051505
11348  064062  047510  042114  042440
11349  064070  041530  042505  042504
11350  064076  006504  000012
11351  064102  052521  051505  044524  ERR206: .ASCIZ  /QUESTIONABLE DRIVE STATUS/<15><12>
11352  064110  047117  041101  042514
11353  064116  042040  044522  042526
11354  064124  051440  040524  052524
11355  064132  006523  000012
11356  064136  047503  052116  047522  ERR207: .ASCIZ  /CONTROLLER TIMED OUT WITH GO SET/<15><12>
11357  064144  046114  051105  052040
11358  064152  046511  042105  047440
11359  064160  052125  053440  052111
11360  064166  020110  047507  051440
11361  064174  052105  005015     000
11362  064201     105  051122  051117  ERR208: .ASCIZ  /ERROR WHILE DIAGNOSING/<15><12>
11363  064206  053440  044510  042514
11364  064214  042040  040511  047107
11365  064222  051517  047111  006507
11366  064230  000012
11367  064232     127                  ERR209: .BYTE   127
11368  064233     110  040505  042504          .ASCIZ  /HEADER ERROR/<15><12>
11369  064240  020122  051105  047522
11370  064246  006522  000012
11371  064252  025052  043052  051111  ERR210: .ASCIZ  /***FIRST ERROR INFO***/<15><12>
11372  064260  052123  042440  051122
11373  064266  051117  044440  043116
11374  064274  025117  025052  005015
11375  064302     000
11376  064303     122  050113  051517  ERR211: .ASCIZ  /RKPOS /
11377  064310  000040
11378  064312  045522  040520  020124  ERR212: .ASCIZ  /RKPAT /
11379  064320     000
11380  064321     101  042104  042522  ERR213: .ASCIZ  /ADDRESS OF BEGINNING OF SECTOR IN ERROR /
11381  064326  051523  047440  020106
11382  064334  042502  044507  047116
11383  064342  047111  020107  043117
11384  064350  051440  041505  047524
11385  064356  020122  047111  042440
11386  064364  051122  051117  000040
11387  064372  005015  040504  040524  ERR214: .ASCIZ  <15><12>/DATA READ/<15><12>
11388  064400  051040  040505  006504
11389  064406  000012
11390  064410  045522  051503  020061  ERR300: .ASCIZ  /RKCS1   RKCS2   RKWCR   RKBA    RKDA    RKDC    RKASOF  RKER/<15><12>
11391  064416  020040  045522  051503
11392  064424  020062  020040  045522
11393  064432  041527  020122  020040
11394  064440  045522  040502  020040
11395  064446  020040  045522  040504
11396  064454  020040  020040  045522
11397  064462  041504  020040  020040
11398  064470  045522  051501  043117
11399  064476  020040  045522  051105
11400  064504  005015     000
11401  064507     122  042113  020123  ERR301: .ASCIZ  /RKDS    RKMR1   RKMR2   RKMR3   RKPOS   RKPAT/<15><12>
11402  064514  020040  051040  046513
```

```
11403   064522  030522  020040  051040
11404   064530  046513  031122  020040
11405   064536  051040  046513  031522
11406   064544  020040  051040  050113
11407   064552  051517  020040  051040
11408   064560  050113  052101  005015
11409   064566     000
11410   064567     122  042113  006523  ERR302: .ASCIZ  /RKDS/<15><12>
11411   064574  000012
11412   064576  005015  042515  051523  ERR303: .ASCIZ  <15><12>/MESS       A       B/<15><12>/ 00     /
11413   064604  020040  020040  020040
11414   064612  040440  020040  020040
11415   064620  020040  041040  005015
11416   064626  030040  020060  020040
11417   064634  020040     000
11418   064637     015  020012  030460  ERR304: .ASCIZ  <15><12>/ 01     /
11419   064644  020040  020040  000040
11420   064652  005015  030440  020060  ERR305: .ASCIZ  <15><12>/ 10     /
11421   064660  020040  020040     000
11422   064665     015  020012  030461  ERR306: .ASCIZ  <15><12>/ 11     /
11423   064672  020040  020040  000040
11424   064700  047125  054105  042520  ERR400: .ASCIZ  /UNEXPECTED MEMORY PARITY TRAP/
11425   064706  052103  042105  046440
11426   064714  046505  051117  020131
11427   064722  040520  044522  054524
11428   064730  052040  040522  000120
11429   064736  044524  042515     072  TIM000: .ASCII  /TIME:/
11430   064743     040  020040  020072  TIM001: .ASCIZ  /   :  :  /<15><12>
11431   064750  035040  020040  005015
11432   064756     000
11433           064760                  .EVEN
```

```
                                        .SBTTL  *** DUMP MEMORY ***

                                        .EVEN
064760  000040                          .BLKW   40              ;RESERVE SPACE FOR STACK
065060  000000                  ..STCK: .WORD   0               ;STORED STACK POINTER

065062  012737  000340  177776  ..DUMP: MOV     #PR7,PS         ;LOCK OUT ALL INTERRUPTS
065070  010637  065060                  MOV     SP,..STCK       ;STORE STACK POINTER
065074  012706  065060                  MOV     #..STCK,SP      ;LOAD STACK POINTER
065100  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
065102  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
065104  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
065106  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
065110  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
065112  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
065114  013705  000224                  MOV     ..LOW,R5        ;STORE FIRST ADDRESS
065120  004737  065452                  JSR     PC,..CR         ;TYPE <CR><LF>
065124  004037  065532                  JSR     R0,..ASCI       ;TYPE R0 =
065130  065640                  ..R0
065132  011646                          MOV     (SP),-(SP)
065134  004737  065556                  JSR     PC,..OCT
065140  004737  065452                  JSR     PC,..CR
065144  004037  065532                  JSR     R0,..ASCI       ;TYPE R1 =
065150  065653                  ..R1
065152  016646  000002                  MOV     2(SP),-(SP)
065156  004737  065556                  JSR     PC,..OCT
065162  004737  065452                  JSR     PC,..CR
065166  004037  065532                  JSR     R0,..ASCI       ;TYPE R2 =
065172  065666                  ..R2
065174  016646  000004                  MOV     4(SP),-(SP)
065200  004737  065556                  JSR     PC,..OCT
065204  004737  065452                  JSR     PC,..CR
065210  004037  065532                  JSR     R0,..ASCI       ;TYPE R3 =
065214  065701                  ..R3
065216  016646  000006                  MOV     6(SP),-(SP)
065222  004737  065556                  JSR     PC,..OCT
065226  004737  065452                  JSR     PC,..CR
065232  004037  065532                  JSR     R0,..ASCI       ;TYPE R4 =
065236  065714                  ..R4
065240  016646  000010                  MOV     10(SP),-(SP)
065244  004737  065556                  JSR     PC,..OCT
065250  004737  065452                  JSR     PC,..CR
065254  004037  065532                  JSR     R0,..ASCI       ;TYPE R5 =
065260  065727                  ..R5
065262  016646  000012                  MOV     12(SP),-(SP)
065266  004737  065556                  JSR     PC,..OCT
065272  004737  065452                  JSR     PC,..CR
065276  004037  065532                  JSR     R0,..ASCI       ;TYPE SP =
065302  065742                  ..SP
065304  013746  065060                  MOV     ..STCK,-(SP)
065310  004737  065556                  JSR     PC,..OCT
065314  004737  065452                  JSR     PC,..CR
065320  004737  065452                  JSR     PC,..CR
065324  004037  065532                  JSR     R0,..ASCI       ;TYPE HEADER
065330  065755                  ..ADD
065332  004737  065452                  JSR     PC,..CR
```

```
11490  065336  004737  065452                  JSR     PC,..CR
11491
11492  065342  010546                  10$:    MOV     R5,-(SP)        ;TYPE ADDRESS
11493  065344  004737  065556                  JSR     PC,..OCT
11494  065350  012703  000004                  MOV     #4,R3           ;TYPE 4 LOCATIONS PER LINE
11495  065354  004037  065532          12$:    JSR     R0,..ASCI
11496  065360  066000                          ..BLNK
11497  065362  012546                          MOV     (R5)+,-(SP)     ;TYPE CONTENTS
11498  065364  004737  065556                  JSR     PC,..OCT
11499  065370  005303                          DEC     R3              ;CHECK IF START OF NEXT LINE
11500  065372  001370                          BNE     12$
11501  065374  004737  065452                  JSR     PC,..CR
11502  065400  005777  113534                  TST     @$WR            ;CHECK IF STOP PRINT OUT
11503  065404  100001                          BPL     15$
11504  065406  000000                          HALT
11505  065410  023705  000226          15$:    CMP     ..HIGH,R5       ;CHECK IF FINISHED
11506  065414  103352                          BHIS    10$
11507  065416  004737  065452                  JSR     PC,..CR
11508  065422  004737  065452                  JSR     PC,..CR
11509  065426  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
11510  065430  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
11511  065432  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
11512  065434  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
11513  065436  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
11514  065440  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
11515  065442  013706  065060                  MOV     ..STCK,SP       ;RESTORE STACK POINTER
11516  065446  000000                          HALT
11517  065450  000776                          BR      .-2
11518
11519  065452  105777  113472          ..CR:   TSTB    @$TPS           ;WAIT FOR READY
11520  065456  100375                          BPL     ..CR
11521  065460  112777  000015  113464          MOVB    #15,@$TPB       ;LOAD <CR>
11522  065466  105777  113456          5$:     TSTB    @$TPS           ;WAIT FOR READY
11523  065472  100375                          BPL     5$
11524  065474  112777  000012  113450          MOVB    #12,@$TPB       ;LOAD <LF>
11525  065502  113701  001155                  MOVB    $FILLS,R1       ;LOAD FILL COUNT
11526  065506  001410                          BEQ     20$             ;RETURN IF NO FILLS
11527  065510  105777  113434          10$:    TSTB    @$TPS           ;WAIT FOR READY
11528  065514  100375                          BPL     10$
11529  065516  113777  001154  113426          MOVB    $NULL,@$TPB     ;LOAD FILL CHARACTER
11530  065524  005301                          DEC     R1              ;CHECK IF FILL COMPLETE
11531  065526  001370                          BNE     10$
11532  065530  000207                  20$:    RTS     PC              ;RETURN
11533
11534  065532  012001                  ..ASCI: MOV     (R0)+,R1        ;STORE ADDRESS OF ASCII
11535
11536  065534  105711                  1$:     TSTB    (R1)            ;CHECK IF FINISHED
11537  065536  001406                          BEQ     20$
11538  065540  105777  113404          2$:     TSTB    @$TPS           ;WAIT FOR READY
11539  065544  100375                          BPL     2$
11540  065546  112177  113400                  MOVB    (R1)+,@$TPB     ;TYPE CHARACTER
11541  065552  000770                          BR      1$
11542
11543  065554  000200                  20$:    RTS     R0              ;RETURN
11544
11545  065556  012701  000006          ..OCT:  MOV     #6,R1           ;LOAD DIGIT COUNT INTO R1
```

```
11546  065562  005002                          CLR     R2              ;CLEAR R2 FOR OUTPUT
11547  065564  000407                          BR      10$             ;GET LEADING DIGIT
11548
11549  065566  005002                  5$:     CLR     R2              ;GET OCTAL DIGIT
11550  065570  006366  000002                  ASL     2(SP)
11551  065574  006102                          ROL     R2
11552  065576  006366  000002                  ASL     2(SP)
11553  065602  006102                          ROL     R2
11554  065604  006366  000002          10$:    ASL     2(SP)
11555  065610  006102                          ROL     R2
11556  065612  052702  000060                  BIS     #60,R2          ;MAKE IT ASCII
11557  065616  105777  113326          15$:    TSTB    @$TPS           ;WAIT FOR READY
11558  065622  100375                          BPL     15$
11559  065624  110277  113322                  MOVB    R2,@$TPB        ;TYPE OCTAL DIGIT
11560  065630  005301                          DEC     R1              ;CHECK IF WHOLE WORD PRINTED
11561  065632  001355                          BNE     5$
11562  065634  012616                          MOV     (SP)+,(SP)      ;ADJUST RETURN
11563  065636  000207                          RTS     PC              ;RETURN
11564
11565  065640  030122  036440  020040  ..R0:   .ASCIZ  /R0 =     /
11566  065646  020040  020040     000
11567  065653     122  020061  020075  ..R1:   .ASCIZ  /R1 =     /
11568  065660  020040  020040  000040
11569  065666  031122  036440  020040  ..R2:   .ASCIZ  /R2 =     /
11570  065674  020040  020040     000
11571  065701     122  020063  020075  ..R3:   .ASCIZ  /R3 =     /
11572  065706  020040  020040  000040
11573  065714  032122  036440  020040  ..R4:   .ASCIZ  /R4 =     /
11574  065722  020040  020040     000
11575  065727     122  020065  020075  ..R5:   .ASCIZ  /R5 =     /
11576  065734  020040  020040  000040
11577  065742  050123  036440  020040  ..SP:   .ASCIZ  /SP =     /
11578  065750  020040  020040     000
11579  065755     101  042104  042522  ..ADD:  .ASCIZ  /ADDRESS   CONTENTS/
11580  065762  051523  020040  041440
11581  065770  047117  042524  052116
11582  065776  000123
11583  066000  020040  020040     000  ..BLNK: .ASCIZ  /    /
11584          066006                  .EVEN
11585          066006                  HDBUFF=.
11586          066212                  BUFADD=.+132.
11587  066006  044124  020105  045522  HELP:   .ASCII  /THE RK06-RK07-K MUST BE PROPERLY FORMATED 22 SECTORS PER TRACK/<15><12>
11588  066014  033060  051055  030113
11589  066022  026467  020113  052515
11590  066030  052123  041040  020105
11591  066036  051120  050117  051105
11592  066044  054514  043040  051117
11593  066052  040515  042524  020104
11594  066060  031062  051440  041505
11595  066066  047524  051522  050040
11596  066074  051105  052040  040522
11597  066102  045503  005015
11598  066106  047524  041040  043505          .ASCII  /TO BEGIN PERFORMANCE TESTING TYPE CONTROL-C <1C>/<15><12>
11599  066114  047111  050040  051105
11600  066122  047506  046522  047101
11601  066130  042503  052040  051505
```

```
11602   066136  044524  043516  052040
11603   066144  050131  020105  047503
11604   066152  052116  047522  026514
11605   066160  020103  057074  037103
11606   066166  005015
11607   066170  042514  040507  020114          .ASCII  /LEGAL COMMANDS ARE:/<15><12><12>
11608   066176  047503  046515  047101
11609   066204  051504  040440  042522
11610   066212  006472  005012
11611   066216  047124  026440  020040          .ASCII  /TN -  TEST DRIVE N/<15><12>
11612   066224  042524  052123  042040
11613   066232  044522  042526  047040
11614   066240  005015
11615   066242  047127  026440  020040          .ASCII  /WN -  WRITE PACK,VERFY PACK, AND TEST DRIVE N/<15><12>
11616   066250  051127  052111  020105
11617   066256  040520  045503  053054
11618   066264  051105  054506  050040
11619   066272  041501  026113  040440
11620   066300  042116  052040  051505
11621   066306  020124  051104  053111
11622   066314  020105  006516      012
11623   066321      120  020116  020055          .ASCII  /PN -  CHANGE PARAMETER & TEST DRIVE N/<15><12>
11624   066326  041440  040510  043516
11625   066334  020105  040520  040522
11626   066342  042515  042524  020122
11627   066350  020046  042524  052123
11628   066356  042040  044522  042526
11629   066364  047040  005015
11630   066370  047123  026440  020040          .ASCII  /SN -  GET STATISTICS ON DRIVE N/<15><12>
11631   066376  042507  020124  052123
11632   066404  052101  051511  044524
11633   066412  051503  047440  020116
11634   066420  051104  053111  020105
11635   066426  006516      012
11636   066431      104  020116  020055          .ASCII  /DN -  DROP DRIVE N AND GET STATISTICS/<15><12><12>
11637   066436  042040  047522  020120
11638   066444  051104  053111  020105
11639   066452  020116  047101  020104
11640   066460  042507  020124  052123
11641   066466  052101  051511  044524
11642   066474  051503  005015      012
11643   066501      116  041440  047101          .ASCII  /N CAN BE 0-7 OR A/<15><12><12>
11644   066506  041040  020105  026460
11645   066514  020067  051117  040440
11646   066522  005015      012
11647   066525      104  052101  020101          .ASCII  /DATA PATTERNS MUST BE WRITTEN BY THE EXERCISER/<15><12>
11648   066532  040520  052124  051105
11649   066540  051516  046440  051525
11650   066546  020124  042502  053440
11651   066554  044522  052124  047105
11652   066562  041040  020131  044124
11653   066570  020105  054105  051105
11654   066576  044503  042523  006522
11655   066604      012
11656   066605      102  043105  051117          .ASCII  /BEFORE PERFORMANCE TESTING/<15><12><12>
11657   066612  020105  042520  043122
```

```
 11658  066620  051117  040515  041516
 11659  066626  020105  042524  052123
 11660  066634  047111  006507  005012
 11661  066642  025052  053452  051101          .ASCII  /***WARNING*** IF PROCESSOR HALT DURING DATA TRANSFER/<15><12>
 11662  066650  044516  043516  025052
 11663  066656  020052  043111  050040
 11664  066664  047522  042503  051523
 11665  066672  051117  044040  046101>
 11666  066700  020124  052504  044522
 11667  066706  043516  042040  052101
 11668  066714  020101  051124  047101
 11669  066722  043123  051105  005015
 11670  066730  020040  020040  052040          .ASCII  /     TO DRIVE, BAD ECC MAY BE WRITTEN ON PACK./<15><12>
 11671  066736  020117  051104  053111
 11672  066744  026105  041040  042101
 11673  066752  042440  041503  046440
 11674  066760  054501  041040  020105
 11675  066766  051127  052111  042524
 11676  066774  020116  047117  050040
 11677  067002  041501  027113  005015
 11678  067010  020040  020040  042040          .ASCIZ  /     DRIVES MUST BE DROPPED BY PROGRAM OR WITH DROP COMMAND/<15><12><12
 11679  067016  044522  042526  020123
 11680  067024  052515  052123  041040
 11681  067032  020105  051104  050117
 11682  067040  042520  020104  054502
 11683  067046  050040  047522  051107
 11684  067054  046501  047440  020122
 11685  067062  044527  044124  042040
 11686  067070  047522  020120  047503
 11687  067076  046515  047101  006504
 11688  067104  005012     000
 11689          000001                  .END
```

```
ABASE = 177440            1174#   1743    1784
ABNORM  025312            1872    4210    4219    5947    5952    6134#   6454
ABNOOO  025400            6149    6153#   6204    6209    6212
ABNOO1  025440            6151    6155    6165#
ACDW1 = 000000            1743    1786
ACDW2 = 000000            1743    1787
ACLO  = 000010            1301#   5944    7878    7910    7953
ACPUOP= 000000            1743    1758
ADDW0 = 000000            1743    1788
ADDW1 = 000000            1743    1789
ADDW10= 000000            1743    1798
ADDW11= 000000            1743    1799
ADDW12= 000000            1743    1800
ADDW13= 000000            1743    1801
ADDW14= 000000            1743    1802
ADDW15= 000000            1743    1803
ADDW2 = 000000            1743    1790
ADDW3 = 000000            1743    1791
ADDW4 = 000000            1743    1792
ADDW5 = 000000            1743    1793
ADDW6 = 000000            1743    1794
ADDW7 = 000000            1743    1795
ADDW8 = 000000            1743    1796
ADDW9 = 000000            1743    1797
ADEVCT= 000000            1743    1749
ADEVM = 000000            1743    1785
AENV  = 000000            1743    1754
AENVM = 000000            1743    1755
AFATAL= 000000            1743    1746
AMADR1= 000000            1743    1771
AMADR2= 000000            1743    1775
AMADR3= 000000            1743    1778
AMADR4= 000000            1743    1781
AMAMS1= 000000            1743    1765
AMAMS2= 000000            1743    1773
AMAMS3= 000000            1743    1776
AMAMS4= 000000            1743    1779
AMSGAD= 000000            1743    1751
AMSGLG= 000000            1743    1752
AMSGTY= 000000            1743    1745
AMTYP1= 000000            1743    1766
AMTYP2= 000000            1743    1774
AMTYP3= 000000            1743    1777
AMTYP4= 000000            1743    1780
APASS = 000000            1743    1748
APRIOR= 000240            1172#   1743
APTCSU= 000040            9981   10694#
APTENV= 000001            9974   10650   10692#
APTSIZ= 000200            3374   10691#
APTSPO= 000100            9976   10652   10693#
ASNORM  013504            4176#   5982    7726
ASN1$   013600            4190#   4208
ASN2$   013632            4189    4196#
ASN3$   013662            4194    4200#   4225    4248    4273    4297    4309    4398    4411    4443    4493
ASN4$   013672            4177    4204#
ASWREG= 000000            1743    1756
```

```
ATESTN= 000000  1743    1747
AUNIT = 000000  1743    1750
AUSWR = 000000  1743    1757
AVAILQ  001314  1810#   3419    3814    4361    4528    5992    6955    7708    8277    8414
AVECT1= 120210  1173#   1743    1782
AVECT2= 000000  1743    1783
A.ABNL  001456  1872#   8676    8847    8864    9058    9078    9555
A.CONT  001460  1873#   8743    8826    8835    8906    8928    8954    8999    9046    9070    9102    9151    9203
                9230    9368    9566    9612    9621    9634
A.NORM  001454  1871#   8796    8893    9014    9562    9642
BAI   = 000020  1260#   9516
BA16  = 000400  1236#   6075    7143
BA17  = 001000  1237#   6075    7143
BDSECT  042356  7331    8345#
BINOCT  052774  3569    6233    6238    6241    6244    6247    6250    6253    6256    6259    6280    6555    6560
                6648    6651    6655    6659    6662    6668    6682    6685    6689    6693    6699    6702    6705
                6711    6832    6835    6841    6844    6847    7425    7429    7603    7629    7632    7636    7645
                7672    7855    7858    9948    10044#  10225
BIT0  = 000001  1157#   1233    1250    1276    1298    1479    1880    4092    4167    4176    4182    5927    6394
                6495    6513    6634    6735    6784    6794    6826    6852    6922    7085    7506    7654    7724
                9619
BIT00 = 000001  1147#   1157
BIT01 = 000002  1146#   1156
BIT02 = 000004  1145#   1155
BIT03 = 000010  1144#   1154
BIT04 = 000020  1143#   1153
BIT05 = 000040  1142#   1152
BIT06 = 000100  1141#   1151
BIT07 = 000200  1140#   1150
BIT08 = 000400  1139#   1149
BIT09 = 001000  1138#   1148
BIT1  = 000002  1156#   1251    1278    1480    1881    4199    4204    4206    6072    6498    6744    6786    6792
                6814    7085    7502    7504    7724
BIT10 = 002000  1137#   1238    1267    1289    1324    1338    1351    1366    1380    1492    1890    4209    7083
                7894    7948
BIT11 = 004000  1136#   1239    1268    1290    1309    1325    1339    1352    1367    1381    1493    4217    7089
                7123    7124    7300    7894    7953    8018    8020    8082    8171
BIT12 = 010000  1135#   1241    1269    1291    1326    1340    1353    1354    1368    1382    1494    7088    7123
                8021    8027    8029
BIT13 = 020000  1134#   1242    1270    1292    1310    1327    1341    1355    1369    1383    1495    7088    7123
                8037    8043    8045
BIT14 = 040000  1133#   1243    1271    1293    1311    1328    1342    1356    1370    1384    1496    1891    8216
                8260    8345    8399
BIT15 = 100000  1132#   1244    1245    1272    1294    1312    1329    1389    1497    1892    6165    6524    6854
                6874    6892    7020    7352    7516    7871    8369    8391    8419
BIT2  = 000004  1155#   1252    1279    1280    1300    1481    1882    4193    4212    4214    6084    6163    6501
                6753    6797    6802    6823    7085    7493    7724    8245    8438
BIT3  = 000010  1154#   1258    1259    1281    1301    1482    1883    4230    4264    4278    4311    4327    5946
                6124    6504    6537    6539    6849    6934    7724    7894    7959    8348
BIT4  = 000020  1153#   1253    1260    1282    1302    1303    1318    1360    1374    1483    1884    3477    4053
                4069    4341    4404    6203    6837    6997    7320    8088    8092    8442    8446    8455
BIT5  = 000040  1152#   1261    1262    1283    1304    1319    1333    1346    1361    1375    1484    1885    4417
                4472    4486    5958    6211    6746    6816    7018    8458    9445
BIT6  = 000100  1151#   1234    1263    1284    1305    1320    1334    1347    1362    1376    1485    1887    4224
                4227    4229    6159    6750    6820    7021    7030    7724    8067    8070    8442
BIT7  = 000200  1150#   1235    1264    1285    1306    1307    1321    1335    1348    1363    1377    1487    4315
```

```
                    4318    5951    6150    6165    6854    6874    6892    6936    6966    6978    7352    7516    7871
                    8369    8391    8419   10201
BIT8   = 000400     1149#   1236    1265    1286    1287    1308    1322    1336    1349    1364    1378    1488    1888
                    6208    7012    7083
BIT9   = 001000     1148#   1237    1266    1288    1323    1337    1350    1365    1379    1489    1889    7083
BLANKS  060530      6239    6245    6251    6257    6283    6565    6649    6652    6656    6660    6665    6683    6686
                    6690    6694    6700    6703    6708    6833    6842    6845    7628    7630    7633    7642    7856
                   11002#
BPTVEC= 000014      1164#
BSBA    001662      1991#   8266    8336*   8353    8383
BSCYLN  001656      1989#   8265    8284    8302*   8311*   8312    8381
BSE    = 000200     1285#   5925    6436    7329    8923    9215
BSSECT  001654      1987#   8264    8283    8301*   8303*   8304    8306*   8326    8382
BSTRAN  042142      8261    8301#   8350
BSTRCK  001655      1988#   8307*   8308    8310*   8318
BSWC    001660      1990#   8262    8340*   8363    8379
BSWORD  001664      1992#   8333*   8370    8374    8394    8397
BUFADD= 066212      3409    8104   11586#
BUFERI  023304      5671#   6008    8238    8372
BUFFAL  022440      3832    4244    4269    4394    5387    5423#
BUFREL  022642      4288    4300    4340    4495    4502    4526    5502#   6022    6758    6900    6921    7006    7682
                    7741    8010    8275    8289    8406    8431    8485
BWAITQ  001320      1811#   3830    3841    4241    4251    4267    4392    4401    4488    4497    5385    5390    5397
B.BA16= 000001      1250#   9510
B.BA17= 000002      1251#   9510
B.CDT  = 000004     1252#   5008    5011    6456    6458    6462    6466    9192    9510
B.CFMT= 000020      1253#   9192    9510    9587
CALSEC  035612      7480    7501    7546#
CALWC   037210      7210    7283    7385    7814#
CCLR   = 100000     1245#   6525    9145
CDT    = 002000     1238#   8899    9490    9527    9577    9597    9928
CERR   = 100000     1244#   6430    8791    8854    8913    9147    9198    9538    9617
CFMT   = 010000     1241#   8899    9490    9527    9577    9597    9928
CHKADD  025032      4275    4328    5960    6000    6070#   8232    8257
CHKCLK  043372      3429    8503#
CINITQ  001324      1812#   3837    4172    4201    5213    5395    5963    5997    6018    6064    6145    6198    6531
                    6972    7027    7092    7261    7303    7316    7326    7355    7461    7489    7542    7713    7738
                    7806    7874    8003    8024    8040    8064    8085    8162    8174    8213    8254    8269    8385
                    8403    8427    9113    9737    9750    9753*   9782    9797    9800*   9933
CLEAR = 000105      1204#
CLKFLG  001374      1837#   3623    8506*   8512*   8521    8559
CLKFRQ  001353      1826#   8533*   8538*   8540*
CLKINT  043464      3771    8521#
CLOCK   043556      8531    8538#
CLR    = 000040     1261#
CMDTO  = 000100     1485#   6136    8670
CMPBUF  023362      5679    5712#   7415
CNTCNT  001630      1974#   3416*   7861*   7862
CNTRL1  027164      5928    6125    6486#
COE    = 001000     1288#   5925    7128    8923    9215
CONCLR= 000176      1222#   9603
CONERR  037322      7039    7045    7051    7097    7103    7130    7139    7154    7162    7169    7403    7842#   8357
                    8365
CONHUN  044032      8578    8580    8582    8590#
CONTRL  026346      1873    6192    6343#
CPUID   001364      1832#   3527*   3737*   5629
```

```
CR     = 000015  1072#  10020  10030  10733  10736
CRLF   = 000200  1073#   3385   9991  10030
CTO    = 004000  1239#   5899   7037   9211
C.INIT   047716  9407#   9892
C.OPT    051740  3793    8947   9018   9114  9830#
C.RTRN   051364  9519    9545   9553   9556  9559   9563  9567  9581  9601  9606  9613  9622  9635
                 9639    9644#
C.SPEC   050446  9447    9523#
DATERR   025424  6009    6162#
DCK    = 100000  1294#   5925   6436   6828  7358   8923  9215
DCLO   = 000020  1303#
DDISP  = 177570  1079#   1726   3362
DDT    = 000400  1308#
DECBIN   054302  3606    3632   4716   4741  4764   4806  4832  10345#
DESL   = 000010  1258#
DETOFF   037026  7249    7449   7533   7776#  8150
DI     = 040000  1243#   8945   9627   9849
DISPLA   001142  1726#   3362*  3370*
DISPRE   000174  1663#   3370
DLT    = 100000  1272#   5911   6434   7334  8881   8917  9213
DLTCNT   001626  1973#   3415*  7342*  7343
DMD    = 000040  1319#
DRA    = 000001  1298#   8813   9548   9860
DRDY   = 000200  1307#   4188   4207   8072
DROP     013004  3819    4084#  4290   4349  4504   5372  6036  6759  6901  6940  7007  7684  7744
                 8011    8433   8486
DROPDV   012260  3896    3974#
DROT   = 000040  1304#   5944   7117
DRPAR  = 000010  1281#   5937   6420   7075
DRPDRV= 002000  1492#   3817   3981   3996  4301   4413  7728  8290  9432  9910
DRVCNT   001632  1975#   3914*  3920*  3924  3976*  3982*  3986  4012*  4018*  4022  4048*  4055*  4059
DRVDSC= 000040  1484#   7903   7972   8467  8831   9006  9075  9107
DRVHRD= 000020  1483#   7903   7965   8464  8838   9049  9220
DRVMSK= 000007  1257#   6184   8759   8774  8932   9418
DRVPDT= 000004  1481#   3468   8941   9011  9023   9805  9870  9884  9887  9914
DRVPOS= 000002  1480#   3468   7023   8941  8975   9469  9741  9805  9870  9914
DRVSTT   056526  4101   10805#
DRVSZD= 010000  1494#   3468   6201   7055  8815   8966  9095  9552  9871
DRVUSE= 000001  1479#   3470   3805   3917  3932   3953  3965  3979  3994  4015  4030  4051  4067
                4168    5949   6142   6148  9432   9910
DRY    = 000200  1306#
DR.CLR= 000005  1229#   6527   8819   8991  9040   9064  9096  9218
DR.SEL= 000001  1228#   8889   8977   9083  9339   9346  9353  9360  9529  9855
DSC    = 040000  1311#
DSWR   = 177570  1078#   1725   3361
DTBAII= 100000  1497#   9432   9514   9910
DTE    = 010000  1291#   5925   6436   7173  9215
DTYE   = 000040  1283#   5925   6390   6436  7887   7924  9215
DTYPE    020410  4538    4993#  5006
D.AWCK= 000000  1405#
D.CERT= 000012  1406#   4730   4944
D.CTRH= 077777  1409#   4822   4948
D.CTRL= 177770  1410#   4823   4949
D.MNCL= 000000  1399#
D.MNSC= 000000  1402#
D.MNTR= 000000  1400#
```

```
D.MXSC= 000025          1403#  4632   4653   4658   4940
D.MXTR= 000002          1401#  4591   4612   4617   4938
D.RATE= 000003          1404#  4685   4942
D.SKET= 000005          1408#  4776   4946
D.UERT= 000005          1407#  4754   4945
D.WTHI= 077777          1411#  4855   4950
D.WTLO= 177770          1412#  4856   4951
ECCCOR  026160          6290#  7407
ECCPAT  001622          1972#  6310*  6311*  6317*  6318*  6323   6324   6330   6331
ECCW  = 020000          1327#
ECH   = 000100          1284#  5925   6436   7395   8923   9215
EC3$    040324          8019   8027#
EC4$    040550          8044   8067#
EC5$    040654          8069   8088#
EC6$    041400          8090   8196#
EC7$    041706          8219   8257#
EMTVEC= 000030          1167#
ERCONT  001564          1957#  3414*  3795   5927*  5954   6072*  6084*  6124*  6140   6194   6495   6498   6501
                        6504   6513   6524*
ERCVY   032002          5977   6169   6978#  8490
ERCVY1  040262          5968   8018#
ERCVY2  043060          6157   8442#
ERRPRO  001566          1964#  3411*  5969   5976*  6166   6168*  6170   6761*  6903*  6942*  6957*  6974*  6994*
                        7009*  7353*  7685*  7696*  7872*  8013*  8270*  8279*  8282*  8388*  8416*  8428*  8435*
                        8491   8497*
ERRPRQ  001330          1813#  5972   6173   6179   8495
ERRVEC= 000004          1160#  3359   3360*  3371*  3387*  3388*  3400*  3401*  3661*  3662*  3664*  3665*  3669*
                        3670*  8503*  8504*  8510*  8516*  8517*  10400  10401  10405*  10406*  10413*  10414*
ERR000  060533          6411   6521   7345   7864   11003#
ERR001  060560          6357   11007#
ERR002  060625          6362   11014#
ERR003  060654          6367   11018#
ERR004  060707          6373   6444   11023#
ERR005  060753          6378   11030#
ERR006  061002          6351   11034#
ERR007  061030          6522   11038#
ERR008  061072          7865   11044#
ERR009  061136          7346   11050#
ERR012  061173          7595   11055#
ERR013  061311          6497   11069#
ERR014  061350          6500   11075#
ERR015  061401          6503   11080#
ERR016  061425          6507   11084#
ERR017  061455          7040   11089#
ERR020  061511          7046   11094#
ERR021  061524          7052   11096#
ERR022  061543          7033   11099#
ERR023  061610          7064   11106#
ERR024  061627          7078   11110#
ERR025  061643          7070   11113#
ERR026  061663          6989   11117#
ERR027  061723          7967   8466   11123#
ERR028  061742          7975   8469   11126#
ERR029  062002          7058   11132#
ERR030  062032          7979   8463   11137#
ERR031  062052          7941   11140#
```

```
ERR032  062066  7944  11142#
ERR033  062103  7912  11145#
ERR034  062112  7915  11147#
ERR035  062125  7926  11149#
ERR036  062142  7931  11152#
ERR037  062155  7936  11154#
ERR038  062201  7112  11158#
ERR039  062216  7120  11161#
ERR040  062233  7297  11165#
ERR042  062243  8187  11168#
ERR043  062262  8131  11172#
ERR044  062277  7098  11176#
ERR045  062315  7104  11179#
ERR046  062330  7222  11181#
ERR047  062350  7339  11185#
ERR048  062364  7390  11187#
ERR050  062376  7469  11190#
ERR051  062415  7131  11194#
ERR052  062435  8075   8079   8448   8460  11197#
ERR053  062513  7950  11206#
ERR054  062554  7956  11213#
ERR056  062622  7921  11221#
ERR057  062635  7961  11223#
ERR058  062676  7971   8472  11229#
ERR059  062753  7000   8457  11237#
ERR060  063024  7422  11244#
ERR062  063050  7510  11248#
ERR064  063075  8168  11252#
ERR065  063105  7312  11255#
ERR066  063130  7404   7851  11259#
ERR067  063150  6830   7853  11262#
ERR070  063171  7140   8366  11265#
ERR071  063214  7155   8358  11269#
ERR072  063240  7163  11273#
ERR073  063261  7170  11276#
ERR074  063306  7015  11280#
ERR100  063351  6673  11287#
ERR101  063462  6641  11301#
ERR102  063544  6644   6678  11310#
ERR108  063552  7669  11311#
ERR111  063563  6839  11313#
ERR200  063611  6384  11317#
ERR201  063651  6392   6396   6439  11323#
ERR202  063721  6742   6809   7513   7985  11330#
ERR203  063757  7691  11336#
ERR204  064012  6939   6949   8432  11341#
ERR205  064047  6899   6929   7683  11346#
ERR206  064102  6748   6818  11351#
ERR207  064136  6406  11356#
ERR208  064201  8445  11362#
ERR209  064232  8193   8452  11367#
ERR210  064252  8478  11371#
ERR211  064303  7423  11376#
ERR212  064312  7427  11378#
ERR213  064321  6550  11380#
ERR214  064372  6556  11387#
```

```
ERR300  064410          6219   6271   11390#
ERR301  064507          6222   11401#
ERR302  064567          6231   11410#
ERR303  064576          6234   11412#
ERR304  064637          6242   11418#
ERR305  064652          6248   11420#
ERR306  064665          6254   11422#
ERR400  064700          9946   11424#
EXAREA  001616          1971#  5065*  5067*  5083   5087*  5103*  5105*  5120   5124*  5126   5128   5129   5133
                        5144   5145*  5146
E.CCLR= 000001          1880#  6354   9150
E.CERR= 001000          1889#  6387   9228
E.CLAT= 000020          1884#  6370   8824   8997   9045   9069   9101
E.CMTO= 040000          1891#  6191   6404
E.CONT  001462          1874#  3413*  6191*  6348   8742*  8824*  8834*  8905*  8927*  8953*  8997*  9045*  9069*
                        9101*  9150*  9202*  9228*  9367*  9565*  9611*  9619*  9632*
E.DLT = 000400          1888#  6381   8905
E.DPAR= 002000          1890#  6415   8834   9367
E.ILLD= 000100          1887#  6376   9611
E.MDS = 100000          1892#  6349   8742   9202   9565
E.NOAT= 000002          1881#  6360   8953
E.SCLR= 000040          1885#  9632
E.UATT= 000004          1882#
E.UDAT= 000010          1883#  6365   8927
E.UNLD= 020000          1495#  3468   6206   8842   8971   9053   9063
E0$     032766          7129   7134#
E1$     033130          7161   7168   7173#
E10$    035310          7466   7493#  8246   8439
E2$     033632          7175   7264#
E3$     034170          7266   7329#
E4$     034326          7335   7358#
E5$     035136          7360   7464#
FBLKC   001670          1997#  3403*  5430   5467*  5515   5540*  5565*
FBLKT   001672          1999#  3404   3409*  3438*  3443*  3446*  3529*  3542*  3545*  3558   3559   3738*  3742*
                        3745*  3746   3747   5432   5514
FLAG    001361          1830#  3333*  3336*  3339*  3342*  3435   3503   3780   10443*
FLPARM  027030          6399   6456#
FMTE  = 000020          1282#  5925   6436   7887   7929   9215
GENERT  021160          3826   5199#
GETBUF  022342          4291   4350   4498   4529   5383#  6066   6760   6902   6941   6956   6973   7008   7686
                        7745   8012   8278   8415   8434   8487
GETCUR  036746          7024   7268   7749#
GETNUM  027074          4237   4456   4549   5344   6466#
GNS   = ******  U       1652   3384   10726  10729  10730
GO    = 000001          1233#  6189
HCRC  = 000400          1286#
HDBUFF= 066006          7323   8117   11585#
HDR.AD  001502          1915#  8875   8879*  9586*
HDR.CT  001504          1916#  8883*  9589*  9592*
HEAD1   001646          1984#  6840   8118*  8123   8126   8165
HEAD2   001650          1985#  6843   8119*  8124   8125   8184
HEAD3   001652          1986#  6846   8120*  8128
HELP    066006          3775   11587#
HLTOOO  021250          6756   6856   6861#  6992   7005   7341   7518   7860   7990   8484
HOLD    027160          4457   6470*  6475*  6481#
HOLD1   027162          5345   6471*  6476*  6482#
```

```
HOUR    001366          1834#  3487*  8555*  8564
HT    = 000011          1070#  9989  10030
HVRC  = 000400          1287#  5925  6436  7264  8923  9215
HZ      001352          1824#  8533  8540
H.HEAD  001507          1919#  7308
IDAE  = 002000          1289#  5937  7887  7939
IE    = 000100          1234#  5379  9155  9223  9641  10194  10210
ILC   = 000001          1276#  5925  6436  7049  9215
ILF   = 000004          1279#
ILLCMD  056042          3905*  5005  10747#
ILLCOM  056011          3906  10742#
INTMSK  001514          1925#  4160  6442  6447  6952  6964  7245  7447  7531  7699  8148  8662  8669
                        8789  8794  8822  8843  8858  8868  8871  8887  8935  8958  8995  9009  9025
                        9043  9054  9067  9076  9099  9110  9419  9540  9550  9560  9864  9882  9897
INTR  = 000300          1224#  4424  5990  6004  9852
IOTVEC= 000020          1165#
IR    = 000100          1263#
I.AERR  046132          8983  8989  9007  9023#  9088  9093  9108
I.ATTN  045420          8753  8771  8911#
I.CCLR  046712          8808  8862  9145#  9221  9544  9605  9868
I.CSTS  047364          5898  8801  9296#
I.CST1  047406          8805  8861  9300#  9535
I.DUAL  046506          8968  9083#
I.ERR   044614          8802  8806#
I.ERRA  044574          8801#  8873
I.ERRC  044602          8764  8792  8804#  8943
I.HDAL  045060          8785  8852#
I.INTR  044266          3448  3763  8732#
I.ISRL  001506          1918#  3434*  4423*  5989*  6003*  8746  8749*  8752*  9156*  9442*  9575*  9640*  9851*
                        9902*
I.ISSU  046774          6528  8820  8890  8978  8992  9041  9065  9084  9097  9187#  9340  9347  9354
                        9361  9856
I.IOO   044432          8750  8768#
I.RTRN  046674          8744  8797  8809  8818  8821  8827  8836  8845  8848  8863  8865  8891  8894
                        8903  8907  8929  8948  8955  8979  8993  9000  9016  9019  9039  9042  9047
                        9056  9059  9066  9071  9079  9085  9098  9103  9116#
I.STAT  047470          8817  9038  9337#  9558
I.STOR  047236          6344  6429  9197  9251#
I.UNLD  046402          8973  9063#
KIPARO= ****** U       10396
LDDATA  023056          3835  4247  4272  4397  4492  5393  5607#
LDEXAR  020506          4906  4910  4914  4918  4922  5050#
LF    = 000012          1071#  10024  10030  10733  10736
LSTCYL  027154          4238  4239  4550  4571  4576  4936  5009*  5012*  5063  5101  6469*  6474*  6479#
MAIN    011356          3782  3783  3790#  3796  3816  3820  3828  3838  3843
MAXBUF  001354          1827#  3583*  3594*  3595  3597*  3608  3751*  3755*  4381  5239
MCLK  = 000400          1322#
MDS   = 001000          1266#  5911  6434  8740  9200  9542
MEMBAS= 172100          1176#  3389
MEMERR  052472          3393  9946#
MEMVEC= 000114          1175#  3393*  3394*
MERD  = 001000          1323#
MESMSK= 000017          1316#
MEWD  = 002000          1324#
MIND  = 000200          1321#
MINUTE  001370          1835#  3486*  8551*  8552  8554*  8565
```

```
MSP    = 000100          1320#
M.CADD= 017760           1392#   7289
M.CDIF= 017760           1391#
M.DRV  = 000007          1388#
M.HEAD= 007000           1395#
M.ID   = 000003          1390#
M.PAR  = 100000          1389#
M.SECT= 000760           1394#   8095
M.SER  = 077770          1393#   4184
NED    = 010000          1269#   5911    6434    7061    8755    8810    9209    9213    9546    9858
NEM    = 004000          1268#   5911    6434    7095    8917    9213
NERSTT  031300           6872#   7079    7121    8080
NEWFLG  027156           4178    4180*   6400*   6451    6480#
NOCHK  = 000400          1488#   9432    9638    9910
NODSC  = 004000          1493#   7903    7968    8470    8988    9092
NORMAL  024222           1871    5898#   6453
NORM1   024616           6007    6012#   7716
NORM2   024654           6010    6013    6015    6021#
NXF    = 000004          1280#   5937    7887    7934
OCTBIN  054146           3514    3577    3656    3689    3712    4546    4559    4588    4600    4629    4641    4670    5061
                         5079    5099    5116   10290#
OCTSTG  053070          10050   10068   10071#
OFFM12= 000260           1423#   7803
OFFM4  = 000220          1421#   7783
OFFM8  = 000240          1422#   7793
OFFP12= 000060           1418#   7798
OFFP4  = 000020          1420#   7778
OFFP8  = 000040          1419#   7788
OFFSET= 000115           1208#   7804    8038    9477
OFST   = 000004          1300#
OPI    = 020000          1292#   5925    6436    7264    8923    9215
OPISTT  031346           6889#   7298    7313    8169    8188
OPRCMD  011614           3872#  10207
OPR000  056050           3884    3939    4001    4037    4075   10748#
OPR001  056065           3876*   3878    3880    3882    3889*   3912    3929    3938*   3948    3961    3974    3991    4000*
                         4010    4027    4036*   4046    4064    4074*  10751#
OPR002  056070           3885   10752#
OPR003  056103           4356    4511   10754#
OPR004  056116           3940    4076   10756#
OPR005  056142           4002    4038   10760#
OPR006  056162           4091   10763#
OPR007  056230           3873   10770#
OPR008  056260           4503    7742   10774#
OPR010  056340           4090    4104    4355    4510    6633    6987   10782#
OPR011  056357           4088*   4089*   4102*   4103*   4353*   4354*   4508*   4509*   6631*   6632*   6985*   6986*  10785#
OPR012  056362           4348   10786#
OPR013  056407           3926    4061   10790#
OPR014  056451           3988    4024   10796#
OPR015  056473           4289   10800#
OPTFLG  001500           1914#   4166*   9207    9837*   9935*
OR     = 000200          1264#
OVLYLD  001365           1833#   3439    3539*   3555*   3739*
O.OVER  001513           1921#   3418*   4160*   6952*   6964*   7245*   7447*   7531*   7699*   8148*   9882
O.RTRN  052444           9853    9857    9869    9872    9895    9935#
O.WAIT  001464           1894#   3432*   6422*   6863    8671    8675*   8768    8787*   8857*   8870*   8885*   9470*   9513*
                         9571*   9585*   9839    9867*   9901*
```

```
PACK   = 000103            1203#   4191    4192
PARCYL   001602            1968#   4548*   4550    4561*   4562    4571*   4576*   4577*   4578    4579
PARM     016170            3952    3964    4533#
PARMDV   012150            3894    3948#
PARM0    001736            1936    2011#
PARM1    002214            1937    2104#
PARM10   004516            2756#   7752    7755    7766    7769    8477
PARM2    002472            1938    2197#
PARM3    002750            1939    2290#
PARM4    003226            1940    2383#
PARM5    003504            1941    2476#
PARM6    003762            1942    2569#
PARM7    004240            1943    2662#
PARSEC   001612            1970#   4631*   4632    4643*   4644    4653*   4658*   4659*   4660    4661
PARSRT   006642            1669    3333#
PARTRK   001606            1969#   4590*   4591    4602*   4603    4612*   4617*   4618*   4619    4620
PAR.EN=  000001            1177#   3392
PAR000   057507            4537   10901#
PAR001   057533            4535*   4536*  10905#
PAR002   057540            4539   10906#
PAR003   057563            4581   10910#
PAR004   057603            4622   10913#
PAR005   057624            4663   10916#
PAR006   057647            4687   10920#
PAR007   057701            4708   10925#
PAR008   057744            4733   10931#
PAR009   060011            4757   10938#
PAR010   060040            4778   10942#
PAR011   060105            4799   10949#
PAR012   060145            4825   10955#
PAR013   060212            4907   10962#
PAR014   060220            4911   10963#
PAR015   060227            4915   10965#
PAR016   060235            4919   10967#
PAR017   060244            4923   10969#
PAR018   060252            4881    5054   10970#
PAR019   060300            4880   10974#
PAR020   060307            5371   10976#
PAR021   060443            4860   10992#
PAR022   060466            4993   10996#
PASCNT   001356            1828#
PAT    = 000020            1318#
PATTAB   004602            1672    2787#   5631    5739    5836    7623
PAT0     004642            2787    2804#
PAT1     004742            2788    2837#
PAT10    005642            2795    3068#
PAT11    005742            2796    3101#
PAT12    006042            2797    3134#
PAT13    006142            2798    3167#
PAT14    006242            2799    3200#
PAT15    006342            2800    3233#
PAT16    006442            2801    3266#
PAT17    006542            2802    3299#
PAT2     005042            2789    2870#
PAT3     005142            2790    2903#
PAT4     005242            2791    2936#
```

```
PAT5     005342            2792    2969#
PAT6     005442            2793    3002#
PAT7     005542            2794    3035#
PBLKT    001524            1936#   3450    3488    3804    3916    3931    3951    3963    3978    3993    4014    4029    4050
                           4066    8667    8762    8937    8965
PBSVAL=  001000            1489#   9372
PCA    = 004000            1325#
PCD    = 010000            1326#
PERINV   001360            1829#   3637*   3647*   3757*   8534    8545    8549
PERIOD   001375            1838#   8534*   8547*   8549*
PGE    = 002000            1267#   5911    6434    7043    9213
PIP    = 020000            1310#
PIRQ   = 177772            1077#
PIRQVE=  000240            1171#
POSCMD   001666            1993#   7022*   8071
PRDSTT   025720            6230#   6749    6819    6991    7987    8474    8480
PRERWD   036024            7431    7514    7591#
PREXAR   017774            4859    4869    4880#   4896
PRGSRT   006676            3334    3337    3340    3344#
PRIBUF   027442            6535#   6851    7517
PRIREG   025652            6215#   6410
PRISER   027616            4105    4357    4512    6575#   6636
PRISTT   026074            6265#   6511    6743    6813    6990    7340    7849    7986    8473    8479
PRITIM   043670            4106    4358    4513    6409    6510    6637    6988    8559#   10442
PRI000   030000            6626#   6732    6803    6999    7325    7507    7843    7909    7918    7947    7964    8444
PRI001   030066            6516    6641#   6755    6825    7989
PRI002   030244            6512    6673#   6752    6822    7003    7515    7850    7988    8475    8481
PRI100   030464            6728#   7014    7032    7057    7063    7069    8074
PRI200   030654            6782#   7077    7111    7119    7221    7296    7311    7389    7468    8078    8130    8167    8186
                           8192
PRSTAT   026132            6220    6223    6272    6279#   6284
PR0    = 000000            1094#
PR1    = 000040            1095#
PR2    = 000100            1096#
PR3    = 000140            1097#
PR4    = 000200            1098#
PR5    = 000240            1099#   1172    1870    3734    3760
PR6    = 000300            1100#   8532
PR7    = 000340            1101#   3344    3388    3394    3662    8504    8563    10438   10439   11440
PS     = 177776            1074#   1075    3344*   3778*   3829*   3834*   3842*   3937*   3941*   3999*   4003*   4035*   4039*
                           4073*   4077*   4084    4085*   4099    4100*   4142*   4246*   4271*   4368*   4391*   4396*   4425*
                           4485*   4491*   5370*   5373*   5383    5384*   5392*   5396*   5400*   6005*   6021*   6023*   6162*
                           8235*   8240*   8244*   8562    8563*   8567*   8653    8657*   8682*   9413    9414*   9644*   9667
                           9668*   9682*   9706    9707*   9716*   9836    9838*   9936*   10206*  10208*  11440*
PSW    = 177776            1075#
PWRSRT   007172            3386#   10445
PWRVEC=  000024            1166#   3355*   3356*   10426*  10437*  10438*
P.ASOF=  000032            1510#   8994*   8995    9035*   9306*   9479*   9480
P.ASSN=  000210            1617#   3472    3477*   3493*   4053*   4069*   4092    4094*   4096*   4147    4163*   4167*   4176
                           4182*   4193*   4199*   4204    4206*   4212    4214*   4224*   4227    4229*   4230*   4264    4278
                           4311    4315    4318*   4327*   4341    4404    4417    4472*   4486    4501*   4507*   4520    5980
                           6394    6634    6735    6932    6934    6936    6966*   7085    7408    7701    7724    8220    8273
                           8348
P.AWCK=  000077            1536#   4695*   4706*   4943*   5263
P.A00  = 000040            1513#   6236    8985*   8986    9002*   9003    9363*   9536*
P.A01  = 000044            1515#   9342*
```

```
P.A10 = 000050   1517#  9349*
P.A11 = 000054   1519#  4183   9356*
P.BAHI= 000007   1472#
P.BALO= 000010   1473#  4281   4435*  5454*  5462*  5676   6080   6701   7145   7151   7180   7254*  7272
                 7323*  7365   7454*  7484*  7538*  7549   7552   7735*  8155*  8266*  8335   8383*  9496
                 9586
P.BAR = 000024   1507#  6081   7145   7152   7478   8355   9032*  9302*
P.BFCP= 000222   1648#  5677*  5678*  5718   7412*
P.BUFF= 000145   1568#  3456*  5457*  5466*  5506   5508*
P.B00 = 000042   1514#  8980*  8981   9364*  9537*
P.B01 = 000046   1516#  9343*
P.B10 = 000052   1518#  7288   9350*
P.B11 = 000056   1520#  7308   8094   8892*  9357*
P.CERT= 000100   1537#  4718*  4730*  4944*  7678
P.CMLG= 000220   1647#  5674*  5719   7413*
P.CMND= 000001   1466#  4169*  4191*  4197*  4222*  4234*  4259*  4308*  4373*  4410*  4421   4437   4439*
                 4442*  5209*  5218*  5220   5290*  5615   5956   5958   5984   5987   5995*  6006   6012
                 6062*  6681   6970*  7022   7025*  7090*  7125*  7251*  7257   7301*  7324*  7451*  7457
                 7481*  7535*  7703   7705   7711*  7731*  7804*  8022*  8038*  8059*  8060   8071*  8083*
                 8152*  8158   8172*  8211*  8225   8227   8236   8248   8297*  8351   8392   8783   9445
                 9449   9461   9471   9477   9483   9486   9492   9500   9507   9523   9569   9583   9603
                 9609   9875   9877   9879
P.CS1 = 000016   1504#  5899   6075   6269   7037   7073   7081   7143   8804*  8860*  8898*  8899*  8901*
                 8902   9027*  9029*  9296*  9299*  9435   9489*  9490*  9492*  9504*  9507*  9509*  9510*
                 9518   9526*  9527*  9529*  9530   9532*  9533   9538   9576*  9577*  9579*  9580   9596*
                 9597*  9599*  9600   9616*  9617   9626*  9627   9917   9927*  9928*  9930*  9931
P.CS1H= 000007   1471#  5008*  5011*  6456   6458*  6462*  6466   8898   9191   9489   9509   9526   9576
                 9587   9596   9927
P.CS2 = 000020   1505#  5911   7043   7061   7067   7095   7101   7334   7464   8810   9030*  9300*  9443*
                 9468   9516*  9517   9525   9542   9546   9573*  9574   9595
P.CYLN= 000002   1467#  4238*  4260*  4306*  4376*  4433*  4444   4446   5210*  5219*  5360*  6120   6684
                 7158   7187   7192   7253*  7280   7322*  7364   7453*  7483*  7537*  7562   7734*  7815
                 8153*  8265*  8295*  8313   8381*  8896   9473   9498   9593   9925
P.DCYL= 000030   1509#  4433   4434   6121   7158   7187   7196   7271   7280   7371   8165   8284*  8302
                 9034*  9308*
P.DPAT= 000121   1545#  4263*  4380*  4484*  5217*  5262*  5622   6710
P.DRVN= 000000   1465#  4088   4102   4159   4353   4508   4535   6185   6423   6631   6951   6963   6985
                 7244   7446   7530   7698   8147   8780   9189   9417   9881
P.DS  = 000036   1512#  4188   4207   4215   4345   4366   5944   6232   6270   7117   7878   7910   7993
                 8072   8813   9037*  9304*  9548
P.DSTT= 000150   1588#  3458*  5200*  5966   6153   6165*  6537   6539*  6740   6786   6792*  6794*  6797
                 6802*  6807   6854*  6872   6874*  6882   6889   6892*  6922   6968*  7020*  7021*  7083
                 7088*  7089*  7123*  7124*  7300*  7320*  7350   7352*  7502   7504*  7506*  7511   7516*
                 7654   7694*  7869   7871*  7983   8018   8020*  8021*  8027   8029*  8037*  8043   8045*
                 8067   8070*  8082*  8088   8092*  8171*  8216   8260*  8345   8369*  8391*  8399*  8419*
                 8442   8446
P.DTS = 000026   1508#  4431   4432   6105   6118   7166   7185   7195   7270   7277   7370   8283*  8301
                 9033*  9303*
P.ECMP= 000117   1543#  4165*  5789*  7414*
P.EPAT= 000062   1522#  6310   6831   7401   7428   7755   7769   7854   8806*  9437   9919
P.EPOS= 000060   1521#  6299   6834   7397   7399   7424   7857   8807*
P.ER  = 000034   1511#  5925   5937   6420*  6828   7049   7075   7109   7128   7173   7264   7329   7358
                 7395   7887   7919   7924   7929   7934   7939   7995   9036*  9305*
P.ERAD= 000217   1646#  5781*  7497   7598   7604   7614   7635   7637
P.ERHD= 000214   1645#  5676*  5717   5723*  5827   5832   7410*  7499   7596   7609   7643
P.ERR = 000212   1633#  3457*  3492*  4087*  4209*  4217*  5201*  5946*  5951*  6150*  6159*  6163*  6203*
```

```
P.EXAR= 000232   1655* 3495  4154  4900  5294
P.HARD= 000200   1611* 4130  6924* 6925
P.IERC= 000120   1544* 4787* 4797* 4947* 6907  6945  7223  7391  7470  7519  7997  8132
P.LCMD= 000135   1562* 4190* 4196* 4221* 4233* 4319* 4369* 4426* 5232* 6642  6647  6663  8422*
P.LCYL= 000136   1563* 4306  4320* 4370* 4427* 5210  5233* 6650  8295  8423*
P.LDPT= 000144   1567* 4429* 5236* 6667
P.LSEC= 000140   1564* 4307  4321* 4371* 4428* 5211  5234* 6658  8296  8424*
P.LTRK= 000141   1565* 6654
P.LWC = 000142   1566* 4322* 4372* 4430* 5235* 6661  8425*
P.MNCL= 000066   1529* 4579* 4937* 5298  5305
P.MNSC= 000074   1533* 4661* 4941* 5322
P.MNTR= 000072   1531* 4620* 4939* 5308
P.MXCD= 000106   1540* 4808* 4811* 4812* 4822* 4823* 4948* 4949* 6027  6032
P.MXCL= 000070   1530* 4578* 4936* 5297  6459* 6463*
P.MXSC= 000075   1534* 4660* 4940* 5321
P.MXTR= 000073   1532* 4619* 4938* 5307
P.MXWT= 000112   1541* 4834* 4837* 4838* 4839* 4840* 4841* 4842* 4843* 4844* 4845* 4855* 4856*
                 4950* 4951* 6046  6051
P.NER = 000206   1614* 4139  6757* 6875* 6876  7004* 7980* 8483*
P.NODR= 000152   1605* 4109  4385* 4477* 4518  5253* 5254* 5276* 5277* 6027  6032
P.NOPI= 000204   1613* 4136  6891* 6893
P.NRD = 000164   1607* 4117  4388* 4389* 4480* 4481* 5281* 5282* 5283* 5287* 5288* 5289* 6040
                 6041
P.NSKI= 000202   1612* 4133  6884* 6894
P.NWRT= 000156   1606* 4113  4386* 4387* 4478* 4479* 5269* 5270* 5271* 5278* 5279* 5280* 6042
                 6044
P.OFST= 000006   1470* 6698  7671  7778* 7783* 7788* 7793* 7798* 7803* 9479
P.PKSR= 000226   1654* 4282* 4283* 4332* 4334* 6598  6600
P.PRST= 000014   1475* 3468* 3470  3491* 3805  3817  3917  3932  3953  3965  3979  3981* 3994
                 3996* 4015  4030  4051  4067  4086* 4168* 4266* 4294  4296* 4301  4375* 4407
                 4409* 4413  4420* 4441* 5222* 5275* 5949  6014  6016* 6136  6142  6148  6201
                 6206  7023* 7055  7259* 7459* 7487* 7728  7732* 7903  7965  7968  7972  7976
                 8062* 8160* 8250  8252* 8267* 8290  8378* 8401* 8461  8464  8467  8470  8670*
                 8799* 8815* 8831* 8838  8842* 8941* 8966  8971  8975* 8988* 9006* 9011  9023*
                 9049  9053* 9063* 9075* 9092* 9095* 9107* 9220* 9372* 9432* 9469* 9493* 9502
                 9508* 9514  9552* 9638  9741  9774  9805* 9870* 9871* 9884  9887* 9910* 9914*
P.QLNK= 000064   1526* 9670* 9675* 9712  9744  9747  9752  9791  9794  9799
P.RATE= 000076   1535* 4343  4682* 4685* 4942* 5257
P.RBAH= 000134   1557*
P.RBAL= 000132   1556* 4331  4435  5453* 5461* 5512  5612  7735  8337
P.RCMD= 000123   1551* 4170* 4190  4192* 4196  4198* 4221  4223* 4233  4235* 4259  4319  4369
                 4374* 4426  5218  5232  5268* 5274* 5286* 5290  6535  6676  6706  7229  7251
                 7393  7435  7451  7481  7535  7731  8059  8137  8152  8211  8422
P.RCYL= 000124   1552* 4239* 4260  4320  4370  4377* 4427  4434* 5219  5233  5306* 5336  5360
                 7734  8423
P.RDPT= 000122   1550* 4151  4263  4429  4482* 4483* 4484  5217  5236  5261*
P.RECT= 000146   1571* 3459* 5199* 6909* 6910  6967* 7693* 7999* 8000  8218  8386  8407
P.RERD= 000147   1572* 6696  7177  7225* 7226  7231  7236  7432* 7433  7437  7442  7472* 7473
                 7521* 7522  7527  7656  7661  7776  7781  7786  7791  7796  7801  8035  8046
                 8048  8050  8052  8054  8056  8058* 8134* 8135  8139  8144  8196  8200  8202
                 8204  8206  8208  8210*
P.RSEC= 000126   1553* 4240* 4261  4321  4323* 4324  4371  4379* 4428  4432* 5223  5234  5334*
                 5361  6960  7733  8420  8424
P.RTRK= 000127   1554* 5320*
```

```
P.RWC = 000130          1555# 4262  4322  4372  4430  4460  4490* 5224  5235  5427* 5509  5613  7736
                        8339  8425
P.SECC= 000176          1610# 4121  7658* 7675
P.SECT= 000004          1468# 4232* 4240  4261* 4307* 4324* 4378* 4431* 4454  5211* 5223* 5361* 6095
                        6692  7166  7185  7191  7252* 7277  7321* 7363  7452* 7482* 7536* 7567  7733*
                        7832  8101  8154* 8264* 8296* 8329  8382* 8897  9474  9499  9594
P.SEEK= 000211          1629# 3825* 3827  4256  4258* 4305* 5206  5208* 5216* 7730* 8030  8294*
P.SERL= 000224          1653# 4183* 4184* 4185* 4186* 4187* 6579
P.SKET= 000104          1539# 4766* 4776* 4946* 6895
P.SMPL= 000116          1542# 4868* 4878* 4952* 5746
P.SOFF= 000174          1609# 4127  7666* 7677
P.SRRD= 000172          1608# 4124  7663* 7676
P.TRCK= 000005          1469# 4448  6108  6688  7561  7823  8321
P.UERT= 000102          1538# 4743* 4754* 4945* 6925
P.WC  = 000012          1474# 4236* 4262* 4381* 4382* 4384* 4386  4388  4390* 4461* 4471* 4475  4490
                        5224# 5252* 5426  5677  6077  6087  6704  7136  7148  7181  7255* 7256* 7273
                        7366  7455* 7456* 7485* 7486* 7539* 7540* 7553  7736* 8156* 8157* 8262* 8374*
                        8379* 8398* 9497
P.WCR = 000022          1506# 6070  7134  7136  8361  9031* 9301*
P0$     016174          4535# 4544  4586  4627  4668  4692  4714  4739  4762  4784  4804  4830  4865
                        4886  4929
P1$     020200          4543  4936#
P10$    020276          4803  4948#
P11$    020312          4829  4950#
P12$    020326          4864  4952#
P2$     020212          4585  4938#
P3$     020224          4626  4940#
P4$     020236          4667  4942#
P5$     020244          4691  4943#
P6$     020250          4713  4944#
P7$     020256          4738  4945#
P8$     020264          4761  4946#
P9$     020272          4783  4947#
QNEWSW= 000000          1192  1418
QUESMK  060515          7606  7639  11000#
Q.INIT= 040000          1496# 3468  9774  9805  9870  9887  9914
Q.POP   051466          3813  5389  5971  6172  8494  9703#
Q.PUSH  051406          3836  3840  4171  4200  4250  4360  4400  4496  4527  5212  5394  5962  5991
                        5996  6017  6063  6144  6178  6197  6530  6954  6971  7026  7091  7260  7302
                        7315  7325  7354  7460  7488  7541  7707  7712  7737  7805  7873  8002  8023
                        8039  8063  8084  8161  8173  8212  8253  8268  8276  8384  8402  8413  8426
                        9112  9665# 9932
Q.RMOV  051626          6134  6486  9774#
Q.SRCH  051542          9737# 9846
RDALHD= 000164          1221# 7324  8783  9583  9875
RDCHR = 104402          10123 10729#
RDDATA= 000121          1210# 4234  4235  5286  5615  5956  6006  6535  6642  6663  6676  6706  7229
                        7393  7435  8137  8236  8351  9877
RDGATE= 100000          1329#
RDHEAD= 000125          1212# 8901  9599
RDLIN = 104403          3874  4974  10730#
RDSTAT= 000141          1220# 4169  4170  7025  9523
RDY   = 000200          1235# 9533  9844
RECAL = 000113          1207# 4222  4223  6970  7090  7125  7301  8083  8172  9486
RELEAS= 000140          1219# 4421  4442  5987  5995  6062  7705  7711  8227  9569
RELFLG  001406          1843#
```

```
RESTRT  006652  1667  3336#
RESTO   007636  3436  3483#
REST1   011236  3724  3735  3762#
REST2   011252  3481  3765#
RESVEC= 000010  1161#
RKASOF= 000016  1188#  8793  8867  8950  9258  9306  9480*
RKBA  = 000004  1183#  9254  9302  9496*
RKBAS   001446  1868#  3666*  3680*  3758*  8658  8738  9424  9841
RKCS1 = 000000  1181#  4424*  5990*  6004*  6188  6427  6525*  6866  8763  8790  8852  8902*  8911
                9145*  9146  9155*  9194*  9195  9223*  9251  9518*  9530*  9532  9580*  9600*  9616
                9626  9641*  9843  9852*  9931*
RKCS2 = 000010  1185#  6183  6425*  6426*  8739  8880  9190*  9252  9300  9468*  9517*  9525*  9574*
                9595*  9615*  9926*
RKDA  = 000006  1184#  8897*  9255  9303  9474*  9499*  9594*
RKDB  = 000024  1191#  8876  8877  8878
RKDC  = 000020  1189#
RKDCYL= 000020  1190#  8896*  9259  9308  9473*  9498*  9593*  9925*
RKDS  = 000012  1186#  9256  9304
RKECPS= 000030  1196#  8807  9263
RKECPT= 000032  1198#  8806  9264
RKER  = 000014  1187#  8920  9257  9305
RKMR1 = 000026  1192#  6424*  6526*  8888*  8976*  9260  9337*  9344*  9351*  9358*  9444*  9924*
RKMR2 = 000034  1193#  9261  9536
RKMR3 = 000036  1194#  9262  9537
RKPAT = 000032  1197#
RKPOS = 000030  1195#
RKPRI   001452  1870#  3449  3723*  3734*  3760*  3764  3777  3829  3937  3999  4035  4073  4085
                4100  4391  4485  5370  5384  5396  6021  6162  8240  8244  8657  9414  9668
                9707  9838  10192  10208
RKVEC   001450  1869#  3447  3693*  3704*  3759*  3762
RKWC  = 000002  1182#  9253  9301  9497*
RLS   = 000010  1259#  9573
RSTAT   036776  7763#  8122
RTYBA   001634  1977#  6294  6552  7180*  7254  7272*  7365*  7410  7454  7478*  7479*  7484  7500*
                7538  7548  7551*  7552*  7838*  8155
RTYCMD  001645  1983#
RTYCYL  001642  1981#  7192*  7196*  7209*  7253  7271*  7294  7322  7364*  7371*  7384*  7453  7483
                7537  7562*  7581*  7814  8153
RTYSCN  001644  1982#  3410*  6920*  7190*  7211*  7219*  7238  7269*  7362*  7585*  7674*  7692*
RTYSEC  001640  1979#  7191*  7195*  7197  7199*  7202*  7240  7252  7270*  7321  7363*  7370*  7372
                7374*  7377*  7452  7482  7536  7575*  7829  8154  8182
RTYTRK  001641  1980#  7203  7205*  7208*  7307  7378  7380*  7383*  7561*  7572*  7577  7584*  7820
                8178
RTYWC   001636  1978#  6295  6553  7181*  7182*  7183  7189*  7212  7214*  7217  7255  7273*  7274*
                7275  7284  7287*  7366*  7367*  7368  7386  7388*  7412  7413  7455  7485  7539
                7560*  7836*  8156
SAVSWR  001404  1842#  10425*  10444
SCLR  = 000040  1262#  6426  9615
SECOND  001372  1836#  3485*  8541*  8542  8544*  8566
SEEK  = 000117  1209#  4308  5209  5984  7703  8022  8225  8297  9461  9471  9930
SELDRV= 000101  1202#  9579
SKI   = 000002  1278#  5937  7109
SKISTT  031332  6882#  7113
SOFCMP  001362  1831#  3610*  3621*  3756*  4858  5674
SPAR  = 020000  1242#  5899  7073  7081  9211
SPDLSS= 000020  1302#  5944  7878  7993
```

```
SRTSPL= 000111          1206#   4197    4198    9483
STACK = 001100          1065#   3351    3790   10440
START   006670          1666    3342#
START$  007262          3402#
STARTI  006662          1674    3339#
STAT    013102          3807    4017    4032    4099#
STATDV  012440          3899    4010#
STATIS  001633          1976#   3417*   3801    3811*   8550*
STATO0  013114          4097    4101#
STATO1  013352          4095    4142#
STKLMT= 177774          1076#
STTO00  056555          6597   10809#
STTO01  056604          4107   10815#
STTO02  056626          4111   10818#
STTO03  056653          4115   10822#
STTO04  056675          4119   10826#
STTO05  056730          4123   10831#
STTO06  056744          4126   10833#
STTO07  056760          4129   10835#
STTO08  057004          4132   10839#
STTO09  057030          4135   10843#
STTO10  057076          4138   10850#
STTO11  057116          6619   10853#
STTO12  057143          6602   10857#
STTO13  057156          6617   10858#
STT1$   031360          6885    6892#
STT100  056600          6578   10813#
STT2$   031404          6877    6897#
STT5$   031444          6873    6878    6883    6890    6896    6898    6907#
SUBCLR= 000177          1223#   9609
SURECY  036276          7417    7654#   8032    8222    8229    8241    8410
SVAL  = 100000          1312#
SWR     001140          1725#   3349    3361*   3363    3369*   3376*   3798    3822    4254    5203    5671    6024    6345
                        6488    6491    6493    6508    6517    6519    6540    6542    6626    6629    6738    6804    6861
                        6897    6927    6947    6980    6983    7001    7337    7420    7508    7667    7680    7689    7844
                        7981    7991    8285    8453   10218   10247*  10425   10444*  11502
SWREG   000176          1664#   3369   10218   10224
SW0   = 000001          1129#
SW00  = 000001          1119#   1129
SW01  = 000002          1118#   1128
SW02  = 000004          1117#   1127
SW03  = 000010          1116#   1126
SW04  = 000020          1115#   1125
SW05  = 000040          1114#   1124
SW06  = 000100          1113#   1123
SW07  = 000200          1112#   1122
SW08  = 000400          1111#   1121
SW09  = 001000          1110#   1120
SW1   = 000002          1128#   5671
SW10  = 002000          1109#   6345    6488    6626    6980
SW11  = 004000          1108#
SW12  = 010000          1107#
SW13  = 020000          1106#   3798    6491    6508    6540    6629    6738    6804    6947    6983    7001    7337    7420
                        7508    7667    7689    7844    7981    8453
SW14  = 040000          1105#   3798    3822    4254    5203    6024    6493    6519    6897    6927    7680    8285
SW15  = 100000          1104#   6517    6861
```

```
SW2    = 000004            1127#
SW3    = 000010            1126#   6542
SW4    = 000020            1125#   6897    6927    7680
SW5    = 000040            1124#   6024
SW6    = 000100            1123#   7991
SW7    = 000200            1122#
SW8    = 000400            1121#
SW9    = 001000            1120#
SYS000   057161            3507   10859#
SYS001   057171            3531   10861#
SYS002   057211            3557   10864#
SYS003   057243            3570   10869#
SYS004   057247            3599   10870#
SYS005   057301            3649   10875#
SYS006   057324            3682   10879#
SYS007   057347            3672   10883#
SYS008   057374            3772   10887#
SYS009   057442            3625   10894#
SYS010   057466            3705   10898#
S.ACLO= 000100              1347#
S.BRHM= 000100              1362#
S.BRKE= 040000              1384#
S.CART= 000400              1364#
S.DCLO= 010000              1354#
S.DIB  = 002000             1380#
S.DOOR= 000200              1363#
S.DRA  = 000040             1333#
S.DROT= 020000              1355#
S.DRY  = 000200             1335#
S.DSC  = 040000             1342#   8829    8986    9003    9073    9090    9105
S.FLT  = 000200             1348#   8981    9086
S.FORM= 001000              1337#
S.FWD  = 002000             1366#
S.HDFL= 000200              1377#
S.HDHM= 000040              1361#
S.ICYL= 000040              1346#
S.ILF  = 000400             1349#
S.LIMD= 020000              1383#
S.LOAD= 010000              1368#
S.MHD  = 000400             1378#
S.NMOV= 010000              1382#
S.OFF  = 002000             1338#
S.PAR  = 001000             1350#   8832    9365
S.PIP  = 020000             1341#   8840    9051
S.PLO  = 004000             1381#
S.REV  = 004000             1367#
S.RTZ  = 020000             1369#
S.SECT= 000020              1374#
S.SKI  = 002000             1351#
S.SPIN= 010000              1340#
S.SPLS= 010000              1353#
S.SPOK= 001000              1365#
S.TYPE= 000400              1336#
S.UNLD= 040000              1370#
S.UNS  = 040000             1356#
S.VV   = 000100             1334#
```

```
S.WCLK= 000040  1375#
S.WGAT= 000100  1376#
S.WLE = 004000  1352#
S.WRL = 004000  1339#
S.XDOK= 000020  1360#
S.XERR= 001000  1379#
S1$     011102  3505    3511    3737#
S2$     011114  3535    3739#
S3$     011144  3574    3743    3746#
S4$     011202  3603    3752    3756#
S5$     011210  3629    3757#
S6$     011214  3653    3758#
S7$     011222  3686    3759#
S8$     011230  3709    3760#
TBITVE= 000014  1162#
TEST    013360  3478    3919    3934    3955    3967    4054    4070    4147#
TESTDV  011774  3892    3912#
TEST1   013434  3474    4165#   4181
TIMHR   001376  1839#   8564*   8570
TIMMIN  001400  1840#   8565*   8579
TIMOUT  025516  6138    6183#
TIMSEC  001402  1841#   8566*   8581
TIMOOO  064736  8584    11429#
TIMOO1  064743  8568    11430#
TKVEC = 000060  1169#   10191*  10192*
TMPCNT  001570  1965#   3567*   3568    3581    3594
TMPSEC  001576  1967#   5321*   5322*   5323*   5327    5330*   5333*   5334    5341
TMPTRK  001572  1966#   5307*   5308*   5309*   5313    5316*   5319*   5320    5337
TPVEC = 000064  1170#
TRAPVE= 000034  1168#   3353*   3354*   3777*   10439*
TRTVEC= 000014  1163#
TSTDEF  020340  3508    3532    3571    3600    3626    3650    3683    3706    4540    4582    4623    4664    4688
                4710    4735    4758    4780    4800    4826    4861    4882    4974#   4994    5055
TSTDRP  037512  7107    7878#
TYPE  = 104401  3382    3507    3521    3523    3531    3550    3552    3557    3570    3589    3591    3599    3616
                3618    3625    3642    3644    3649    3672    3675    3677    3682    3699    3701    3705    3729
                3731    3768    3772    3775    3873    3884    3885    3906    3926    3939    3940    3988    4001
                4002    4024    4037    4038    4061    4075    4076    4090    4091    4101    4104    4107    4111
                4115    4119    4123    4126    4129    4132    4135    4138    4141    4289    4348    4355    4356
                4359    4503    4510    4511    4514    4537    4539    4566    4568    4581    4607    4609    4622
                4648    4650    4663    4677    4679    4687    4701    4703    4708    4725    4727    4733    4749
                4751    4757    4771    4773    4778    4792    4794    4799    4817    4819    4825    4850    4852
                4860    4873    4875    4880    4881    4893    4895    4993    5005    5052    5054    5152    5154
                5371    6219    6222    6231    6234    6239    6242    6245    6248    6251    6254    6257    6260
                6271    6283    6286    6347    6351    6357    6362    6367    6373    6378    6384    6392    6396
                6406    6408    6411    6439    6444    6490    6497    6500    6503    6507    6521    6522    6550
                6556    6558    6565    6568    6597    6619    6628    6633    6641    6644    6645    6649    6652
                6656    6660    6665    6669    6673    6678    6679    6683    6686    6690    6694    6700    6703
                6708    6712    6733    6737    6742    6748    6809    6810    6812    6818    6830    6833    6836
                6839    6842    6845    6848    6899    6929    6939    6949    6982    6987    6989    7000    7339
                7345    7346    7422    7423    7426    7427    7430    7510    7513    7595    7606    7628    7630
                7633    7639    7642    7646    7669    7673    7683    7691    7742    7846    7848    7853    7856
                7859    7864    7865    7912    7915    7921    7926    7931    7936    7941    7944    7950    7956
                7958    7961    7967    7971    7975    7979    7985    8432    8445    8448    8449    8452    8457
                8460    8463    8466    8469    8472    8478    8584    9946    9949    9994    10068   10130   10136
                10141   10145   10150   10151   10153   10156   10160   10204   10215   10222   10223   10226   10238
```

```
                    10249   10266   10441   10614   10726#
T.ASOF  001424       1855#   6442    8793*   8794    8822    8867*   8868    8950*   8951    8958    8994    9035    9043
                     9067    9099    9258*
T.BA    001416       1852#   9032    9254*
T.CONT= 000012       1428#   7862
T.CS1   001410       1847#   6188*   6189    6217    6430    6527*   8763*   8790*   8791    8804    8819*   8852*   8854
                     8860    8889*   8911*   8913    8945    8977*   8991*   9028*   9029    9040*   9064*   9083*   9096*
                     9146*   9147    9187    9191*   9192*   9194    9198    9211    9251*   9298*   9299    9339*   9346*
                     9353*   9360*   9843*   9844    9849    9855*
T.CS2   001412       1849#   6434    8739*   8740    8755    8758    8773    8880*   8881    8917    8931    9030    9188*
                     9189*   9190    9200    9209    9213    9252*   9858    9862
T.DA    001420       1853#   9033    9255*
T.DB    001444       1864#   6221
T.DC    001422       1854#   9034    9259*
T.DLT = 001000       1427#   7343
T.DS    001430       1858#   6218    9037    9256*   9860
T.ER    001426       1857#   6390    6436    8920*   8923    9036    9215    9257*
T.MR1   001432       1859#   9260*
T.MR2   001434       1860#   8829    8840    8985    9002    9051    9073    9090    9105    9261*   9342    9349    9356
                     9363
T.MR3   001436       1861#   8832    8892    8980    9086    9262*   9343    9350    9357    9364    9365
T.NER = 000020       1414#   6876
T.PAT   001442       1863#   9264*
T.POS   001440       1862#   9263*
T.WCR   001414       1851#   9031    9253*
UEXATT= 000010       1482#   7903    7976    8461    8799
UFE   = 000400       1265#   5911    6434    7067    8755    8810    9213    9546
UNLOAD= 000107       1205#   9449
UNRECY  031502       6915    6920#   7233    7439    7475    7524    8006    8141
UNS   = 040000       1293#   5937    7887    7919    7995
UPE   = 020000       1270#   5911    6434    7101    6917    9213
VV    = 000100       1305#   4215
WCE   = 040000       1271#   5911    7464    8917    9213
WLE   = 004000       1290#   5937    7887    8923
WRDATA= 000123       1211#   5268    9504
WRHEAD= 000127       1213#
WRL   = 004000       1309#   4345    4366
WRTCHK= 000131       1214#   4373    4374    4410    4437    4439    5220    5274    6012    7257    7457    8060    8158
                     8248    8392    9500    9879
WRTEPK  012614       3786    3903    4046#
WRTGAT= 040000       1328#
WRTVER  015234       4313    4404#
WVOS    015276       4408    4413#   8287
WV1S    016040       4303    4503#   8292
WV2S    016052       4419    4507#
WV5S    016162       4505    4529#
WV9S    016166       4530#
W.DRV   001544       1947#   3424    6449*   8664*   8788*   8844*   8859*   8872*   8886*   8936*   9010*   9026*   9055*
                     9077*   9111*   9422*   9485*   9488*   9541*   9551*   9561*   9866*   9900*
W.MILI  001470       1897#   3431    8656    9624
W.MIN   001476       1913#   9485    9551    9866
W.MTIM  001466       1896#   3431*   8654*   8656*
W.SEC   001472       1910#   9422    9900
W.TIME  001512       1920#   3430*   6447*   8662    8669*   8789*   8843*   8858*   8871*   8887*   8935*   9009*   9025*
                     9054*   9076*   9110*   9419*   9540*   9550*   9560*   9864*   9897*
W.WCK = 000200       1487#   4266    4294    4296    4375    4407    4409    4420    4441    5222    5275    6014    6016
```

```
                        7259    7459    7487    7732    8062    8160    8250    8252    8267    8378    8401    9432    9493
                        9502    9508    9910
W.WTCH  044112           3794    8649#   9531
W.BSEC  001474           1912#   8844    9055    9488
XXDPMG  055730           3768   10733#
$APTHD  001000           1685    1691#
$ASTAT= ****** U        10672   10687
$ATYC   055442          10643   10645#
$ATY1   055416          10641#
$ATY3   055424           9979   10642#
$ATY4   055434          10644#
$AUTOB  001134           1722#  10220
$BASE   001244           1784#
$BDADR  001122           1717#
$BDDAT  001126           1719#
$BELL   001160           1735#   6347    6490    6628    6982
$B2D    055236          10572   10588#
$CDW1   001250           1786#
$CDW2   001252           1787#
$CHARC  052770           9996*  10006*  10013   10022*  10027#
$CKSWR= ****** U        10729
$CMTAG  001100           1705#   3346    3347
$CM3  = 000000           1735#
$CNTLC  054141          10204   10271#
$CNTLG  053512          10171#  10222
$CNTLU  053505          10145   10170#  10238
$CPUOP  001216           1758#
$CRLF   001165           1737#   4141    4359    4514    6260    6286    6408    6558    6568    6645    6669    6679    6712
                        6737    6812    6836    6848    7426    7430    7646    7673    7848    7859    7958    8449    9949
                        9995   10030   10150   10170   10249
$DB2D   055174           4110    4114    4118   10574#
$DDW0   001254           1788#
$DDW1   001256           1789#
$DDW10  001300           1798#
$DDW11  001302           1799#
$DDW12  001304           1800#
$DDW13  001306           1801#
$DDW14  001310           1802#
$DDW15  001312           1803#
$DDW2   001260           1790#
$DDW3   001262           1791#
$DDW4   001264           1792#
$DDW5   001266           1793#
$DDW6   001270           1794#
$DDW7   001272           1795#
$DDW8   001274           1796#
$DDW9   001276           1797#
$DECNT  055300          10569*  10584*  10604#
$DECVL  055402          10588   10614   10637#
$DEVCT  001200           1749#
$DEVM   001246           1785#
$ENV    001210           1754#   9974   10650   10674
$ENVM   001211           1755#   3374    9976    9981   10652
$ERFLG  001103           1708#
$ERMAX  001115           1714#
$ERRPC  001116           1715#
```

```
$ERRTB  001314          1808#
$ERTTL  001112          1712#
$ETABL  001210          1753#
$ETEND  001314          1697    1806#
$FATAL  001172          1746#  10678#
$FFLG   055662         10641*  10644*  10672   10681*  10689#
$FILLC  001156          1733#   9999   10030
$FILLS  001155          1732#  10030   11525
$GDADR  001120          1716#
$GDDAT  001124          1718#
$GTSWR= ****** U       10728
$HD    = 000000         1031
$HIBTS  001000          1692#
$HINUM  055126          3483*   5255    5259    5265    5312    5326    10522   10530   10535*  10540#
$HIOCT  054300         10315*  10328#
$ICNT   001104          1709#
$INTAG  001135          1723#
$ITEMB  001114          1713#
$LF     001166          1738#  10030   10160   10170
$LFLG   055661         10682*  10688#
$LKCSB  001336          1818#   8505    8525*
$LKCSR  001334          1817#   8526*
$LKS    001342          1820#   8511    8530*
$LLVEC  001344          1821#   8529
$LONUM  055130          3484*   5238    5301    10521   10528   10534*  10541#
$LPADR  001106          1710#
$LPERR  001110          1711#
$LPVEC  001340          1819#   8524
$LSCS   001346          1822#
$LSDB   001350          1823#
$LSTAD  054550          3438    3529    3738    10415*  10419#
$MADR1  001222          1771#
$MADR2  001226          1775#
$MADR3  001232          1778#
$MADR4  001236          1781#
$MAIL   001170          1693    1697    1744#   3373    9974
$MAMS1  001220          1765#
$MAMS2  001224          1773#
$MAMS3  001230          1776#
$MAMS4  001234          1779#
$MBADR  001002          1693#
$MFLG   055660         10642*  10648   10683*  10687#
$MNEW   053530         10174#  10226
$MSGAD  001204          1751#  10658*  10661
$MSGLG  001206          1752#  10663*
$MSGTY  001170          1745#  10656   10664*  10676   10680*
$MSWR   053517         10172#  10223
$MTYP1  001221          1766#
$MTYP2  001225          1774#
$MTYP3  001231          1777#
$MTYP4  001235          1780#
$MULT   054714          4451    5085    5122    5135    5241    5302    5315    5329    5339    7826    8324    10470#
$NULL   001154          1731#  10001   10030   11529
$PASS   001176          1748#   3373*
$PASTM  001006          1695#
$POWER  054702         10441   10448#
```

```
$PWRCT  054676          10431* 10432* 10433* 10435* 10447#
$PWRDN  054552           3355  10425# 10437
$PWRUP  054572          10426  10431#
$QUES   001164           1736#  3523   3552   3591   3618   3644   3677   3701   3731   4568   4609   4650   4679
                         4703   4727   4751   4773   4794   4819   4852   4875   4895   5154  10030  10153  10170
                        10266
$RAND   055030           5237   5296  10517#
$RDCHR  053100          10090# 10729
$RDDEC= ****** U        10731
$RDLIN  053220          10118# 10730
$RDOCT= ****** U        10731
$RDSZ = 000031          10111#
$R2A  = ****** U        10731
$SAVRE= ****** U        10731
$S82D   055132           4122   4125   4128   4131   4134   4137   4140  10560#
$SETUP= 000014           3331#  3352   3353   3355   3357   3382   3383  10079  10176
$SIZE   054454           3402  10398#
$STUP = 177777           3331#
$SWR  = 162000           1018#  1031   1050   1051   1052   1053   1054   1735
$SWREG  001212           1756#  3376
$TESTN  001174           1747#
$TKB    001146           1728#  10077  10094  10100  10193  10199  10209  10233
$TKINT  053542           3776  10191#
$TKS    001144           1727#  5227#  5379#  10077  10092  10098  10194#  10205#  10210#  10227#  10230  10250#
$TKSRV  053572          10191  10199#
$TN   = 000001           1018#  1031
$TNPW   055356          10570  10571  10627#
$TNPWR  055332          10585  10586  10617#
$TPB    001152           1730#  10019#  10030  11521#  11524#  11529#  11540#  11559#
$TPFLG  001157           1734#  9968   10030
$TPS    001150           1729#  10017  10030  11519  11522  11527  11538  11557
$TRAP   055664           3353  10703#
$TRAP2  055706          10714# 10725
$TRP  = 000004          10718# 10727#  10729  10730#  10731#
$TRPAD  055720          10708  10725#
$TSTM   001004           1694#
$TSTNM  001102           1707#
$TTYIN  053454          10120  10121  10133  10151  10165  10169#
$TYPBN= ****** U        10727
$TYPDS= ****** U        10727
$TYPE   052512           9968#  10669  10718  10726
$TYPEC  052724           9998   10005  10012  10017#  10018  10252
$TYPEX  052772          10023  10025  10028#
$TYPOC= ****** U        10727
$UNIT   001202           1750#
$UNITM  001010           1696#
$USWR   001214           1757#
$VECT1  001240           1782#
$VECT2  001242           1783#
$.$.SM= ****** U         1896   3402   3765   8687   8903   9146   9157   9195   9519   9531   9581   9601   9616
                         9643   9932   9944
$2DEC   055242          10589# 10605
.     = 067107           1658#  1662#  1668#  1673#  1675#  1681   1682#  1684#  1686#  1704#  1739   3350   3385#
                         6413   7348   7867  10030  10071#  10077  10169#  10170  10176#  10428  10637#  10690#  10813#
                        10857# 11433# 11437# 11439  11517  11584#  11585  11586
.$ASTA= ****** U        10642  10645
```

```
.$X    = 001000          1681#  1686
..ADD    065755         11488  11579#
..ASCI   065532         11451  11456  11461  11466  11471  11476  11481  11487  11495  11534#
..BLNK   066000         11496  11583#
..CR     065452         11450  11455  11460  11465  11470  11475  11480  11485  11486  11489  11490  11501  11507
                        11508  11519# 11520
..DUMP   065062          1670  11440#
..HIGH   000226          1672#  11505
..LOW    000224          1671#  11439  11449
..OCT    065556         11454  11459  11464  11469  11474  11479  11484  11493  11498  11545#
..R0     065640         11452  11565#
..R1     065653         11457  11567#
..R2     065666         11462  11569#
..R3     065701         11467  11571#
..R4     065714         11472  11573#
..R5     065727         11477  11575#
..SP     065742         11482  11577#
..STCK   065060         11438# 11441* 11442  11483  11515
```

```
COMMEN  1172#
DEFLT   1656#  3508   3532   3571   3600   3626   3650   3683   3706   4540   4582   4623   4664   4688   4710
        4735   4758   4780   4800   4826   4861   4882   5055
ENDCOM  1172#
ERROR   1066#
ESCAPE  1172#
GETPRI  1172#  10404
GETSWR  1172#   3383#
MULT    1172#
M.DUMP     1#  11434
NEWTST  1172#
PARMBK     1#   2008   2101   2194   2287   2380   2473   2566   2659   2753
PARMDF     1#   1429
PARMS   1656#  2040   2133   2226   2319   2412   2505   2598   2691
POP     1172#  10064  10497  10536  10607  10684  10685  11509
PUSH    1172#  10044  10470  10517  10560  10574  10645  10647  10668  11443
P.DRVR     1#   1844
QINPUT  1656#  3519   3548   3587   3614   3640   3673   3697   3727   4564   4605   4646   4675   4699   4723
        4747   4769   4790   4815   4848   4871   4891
REGDEF     1#   1178
REPORT  1172#
R.DRIV     1#   8607
R.QDRV     1#   9652
SCANST  1656#  4554   4595   4636   5070   5090   5107
SCOPE   1067#
SETPRI  1172#
SETTRA 10718#  10729  10730
SETUP   1172#   3345
SKIP    1172#
SLASH   1172#
SPACE   1172#
STARS   1032   1045   1172#  1678   1680   1687   1700   1739   1742   3846   3870   4959   4972   5018   5048
        5162   5197   5404   5421   5482   5500   5575   5605   5689   5710   5875   5896   6716   6726   6766
        6780   8616   8647   8690   8730   9126   9143   9161   9185   9236   9249   9268   9294   9313   9335
        9378   9405   9654   9663   9688   9701   9722   9735   9761   9772   9814   9828   9953  10033  10042
       10076  10082  10111  10180  10189  10275  10288  10331  10343  10392  10420  10454  10468  10508  10545
       10558  10640  10697
SWRSU   1172#   3357#
S.DRVE     1#   1330
TRMTRP 10718#
TYPBIN  1172#
TYPDEC  1172#
TYPNUM  1172#   3378
TYPNUM  1172#
TYPOCS  1172#
TYPOCT  1172#
TYPTXT  1172#
$DECBN     1#  10329
$OCTBN     1#  10273
$$CMRE  1698#
$$CMTM  1698#
$$ESCA  1172#
$$NEWT  1172#
$$SET  10718#  10729  10730
$$SETM  3373#
$$SKIP  1172#
```

```
.EQUAT   1018#   1062
.HEADE   1018#   1021
.SETUP   1018#   3331
.SWRHI   1018#   1046
.SWRLO   1018#   1054#   1056    1058    1060    1061
.$APTB   1740#
.$APTH   1018#   1676
.$APTY   1018#  10638
.$CATC   1018#   1656
.$CMTA   1018#   1698
.$READ   1018#  10074
.$SIZE   1018#  10390
.$TRAP   1018#  10695
.$TYPE   1018#   9951

. ABS.   067107     000

 ERRORS DETECTED:  0

 RM03:CZR6PC,RM03:CZR6PC.SEQ/SOL/CRF/NL:TOC/DOC/EQ:ONEWSW=RM03:DRIV11.P11,RM03:DUMP.P11,RM03:CZR6PC.P11
 RUN-TIME:  31  32  3 SECONDS
 RUN-TIME RATIO:  2044/67=30.4
 CORE USED:  42K  (83 PAGES)

 DOCUMENT PAGES:  280
```